

JOHNSON GRANT 8

IN-61-CR

308246

P-130

IDSE Version I User's Manual

Richard Mayer

Texas A & M University

November 1988

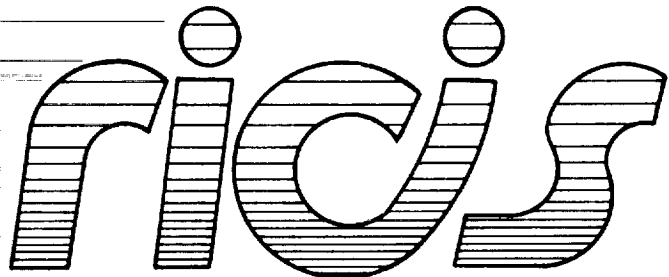
**Cooperative Agreement NCC 9-16
Research Activity IM.6**

(NASA-CR-187255) IDSE VERSION 1 USER'S
MANUAL (Houston Univ.) 130 p CSCL 09B

N91-13085

Unclas

63/61 0308246



**Research Institute for Computing and Information Systems
University of Houston - Clear Lake**

T · E · C · H · N · I · C · A · L R · E · P · O · R · T

The RICIS Concept

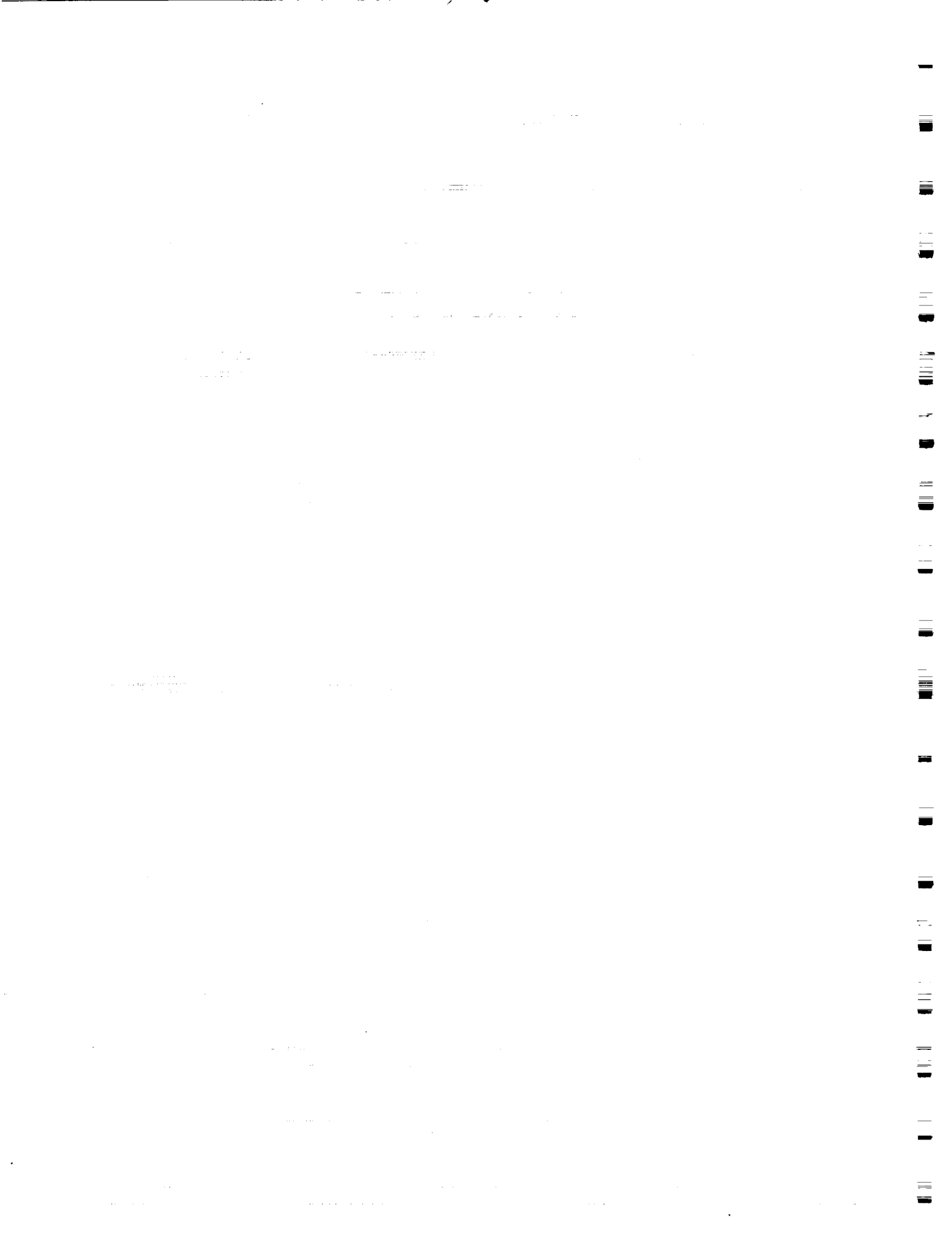
The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

***IDSE Version I
User's Manual***

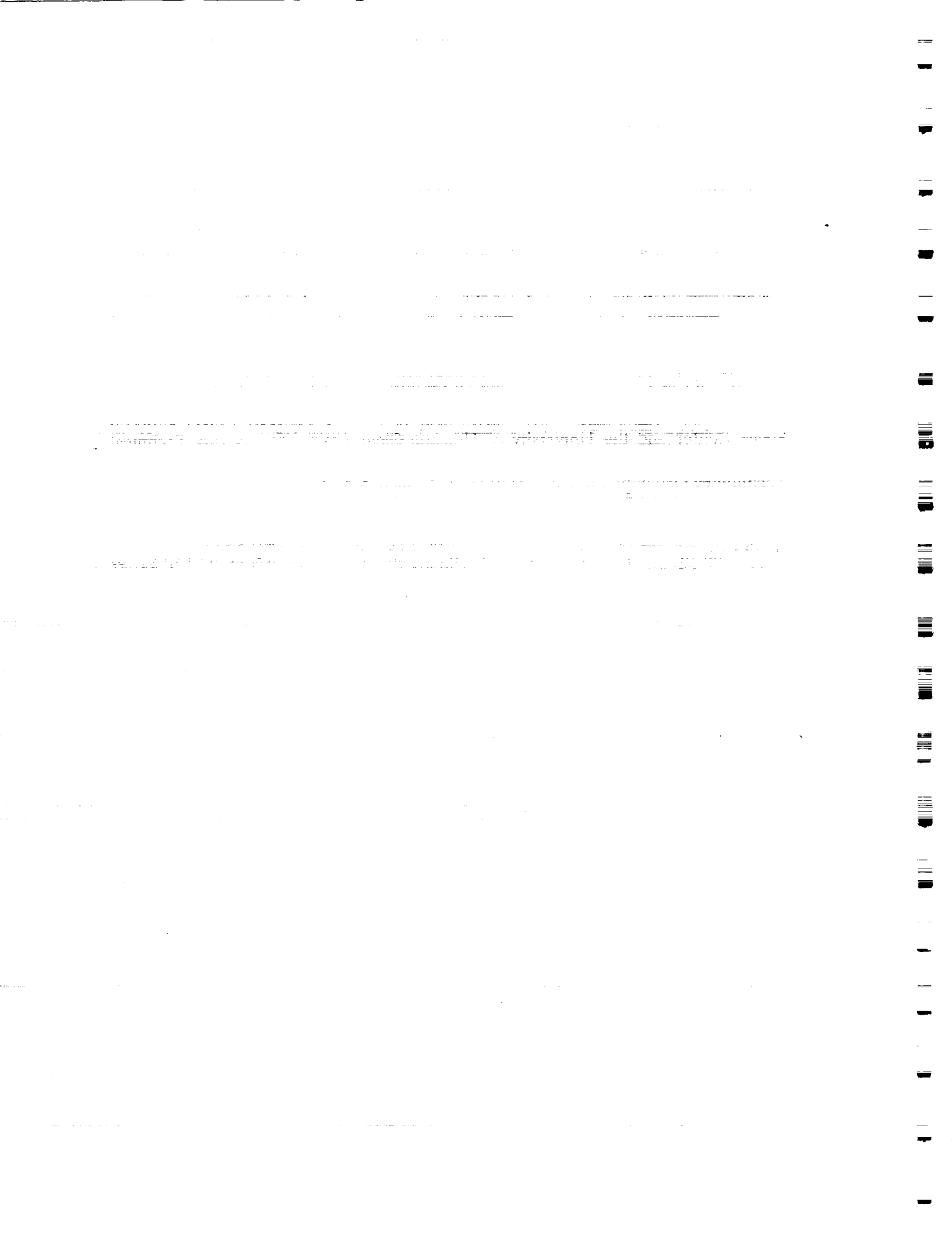


Preface

This research was conducted under the auspices of the Research Institute for Computing and Information Systems by Richard Mayer, Director, Knowledge Based Systems Lab, Department of Industrial Engineering, Texas A & M University. Peter Bishop, Director of the Space Business Research Center at the University of Houston-Clear Lake, served as the technical representative for RICIS.

Funding has been provided by the Mission Planning and Analysis Division, NASA/JSC through cooperative Agreement NCC 9-16 between NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA Technical Monitor for this activity was Robert Savely, Head, Artificial Intelligence Section, Technology Development and Applications Branch, NASA/JSC

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of NASA or the United States Government.



IDSE Version I User's Manual

**Knowledge Based Systems Laboratory
Department of Industrial Engineering
Texas A & M University
College Station, Tx 77843
(409) 845-9363**

IDSE Version 1 User's Manual

November 1988

Prepared by:

Knowledge Based Systems Laboratory
Texas Engineering Experiment Station
Texas A&M University
College Station, TX 77843
(409) 845-9363

NOTE: This document details the pre-release Model Builder System. Known bugs exist and only a portion of the planned functionality is available for IDEF0 and IDEF1.

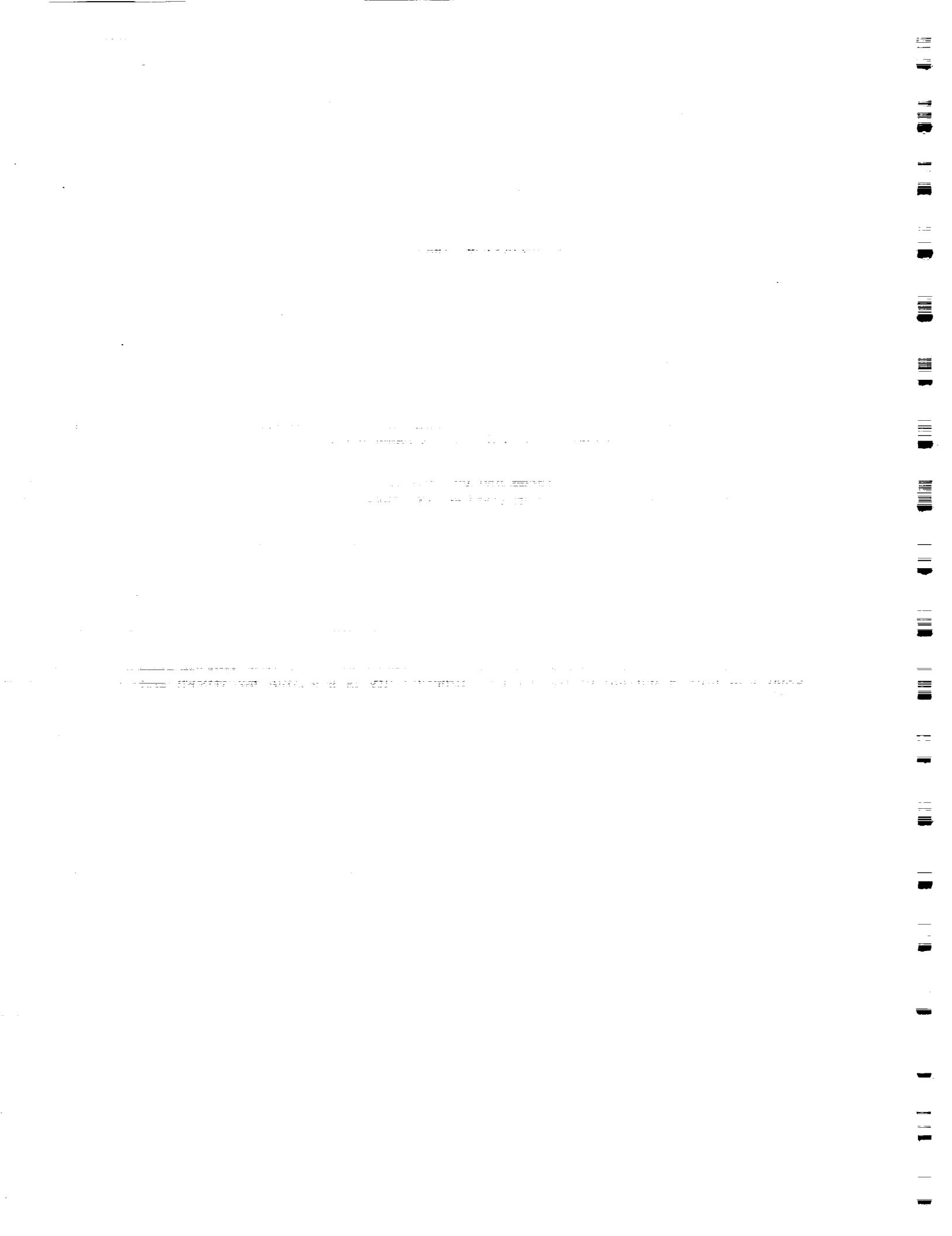


Table of Contents

	Page
1 MDSE Modeling Tools Requirements and Design	3
1.1 The IDEF0 Modeler	3
1.2 IDEF0 Concept Editor	6
1.3 IDEF0 Node Editor	7
1.4 IDEF1 Modeler	8
1.4.1 IDEF1 Tool Design	9
1.4.2 Features of the IDEF1 Tool	9
1.5 IDEF1 Data Dictionary Editor	10
1.5.1 Project Data Dictionary	10
1.5.2 Project Data Dictionary Tool	10
2 MDSE Model Development Support Requirements and Design	12
2.1 Model Validation and Support	13
2.2 Model Integration Support	14
2.3 Model Application / Analysis Support	15
2.4 Architectural Issues	15
2.5 IDEF Model Builder Windows	15
2.6 The Temporary Windows	17
2.7 IDEF Model Builder Status Line and Mouse Status Line	18
3 Getting Started on the MDSE	20
3.1 Running IDEF0 and IDEF1	20
3.2 Sample Usage Scenario for IDEF0	23
3.2.1 Creating a Model, Activities, Concepts, and Relations	23
3.2.2 Hints and Assumptions of IDEF0	30
3.3 Sample Scenario for IDEF1	32
4 MDSE Commands	43
4.1 Command Summary	43
4.2 General Model Builder Commands	47
4.3 IDEF0 Modeler Command Reference	50
4.4 IDEF0 Concept Editor Commands	52
4.5 IDEF0 Node Tree Editor Commands	53
4.6 IDEF1 Modeler Command Reference	55
4.7 IDEF1 Data Dictionary Editor Commands	61
5 Model Element Dictionary Development User's Manual	65
5.1 MEDD Development Platform	65
5.2 MEDD Architecture	65

5.3	Getting Started with the MEDD	66
5.4	Summary of Commands for the MEDD	68
6	Micro-Based Fact Collection Tool User's Manual	73
6.1	Getting Started	74
6.1.1	Main Screen	74
6.1.2	Data Collection	74
6.1.3	Setting the Stage	74
6.1.4	Entering Dialogue Statements	75
6.1.5	Entering Detailed Descriptions	75
6.1.6	Analysis Mode Overview	76
6.1.7	Model Element Dictionary Development Mode	76
6.2	Summary of Commands	77
6.2.1	Keyboard Functions	77
6.2.2	Main Screen Commands	77
6.2.3	Collection Screen Commands	77
6.2.4	Analysis Screen Commands	78
6.2.5	Modeling Screen Commands	79
6.2.6	Dialogue Window Commands	79
6.2.7	Item Description Form Commands	81
A	Installation and Hardware and Software Requirements for the MDSE	84
1.1	IDEF0 and IDEF1 Hardware and Software Requirements	84
1.2	IDEF0 and IDEF1 Installation Procedures	84
B	Installation and Hardware and Software Requirements for the MEDD	86
2.1	Setting up Directories	86
2.2	Modifying AUTOEXC.BAT and CONFIG.SYS	87
2.3	Invoking MEDD Via Q-PRO4	88
2.4	Hardware and Software Requirements	88
2.4.1	Computers	88
2.4.2	Displays	88
2.4.3	Display Adapters	89
2.5	Other Requirements	89
2.6	Distribution and Support of the MEDD	89
C	Installation and Hardware and Software Requirements for the FCT	91
3.1	Floppy Disk Installation	91
3.2	Hard Disk Installation	91
3.3	Technical Information	92
D	Introduction to Symbolics	93
4.1	Conventions and Definitions	93
4.1.1	Screen Configuration	95
4.2	Keys to Become Familiar with	95

4.3	Logging in and Out	97
4.3.1	Logging In	97
4.3.2	Logging Out	98
4.4	Powering up the Machine	99
4.5	Cold-Booting the Machine	99
4.6	Creating a User ID	101
4.7	Getting Around in the Symbolics World	103
4.8	Machine Activities	104

List of Figures

	Page
1 IDEF0 Model Development Diagram	4
2 IDEF1 Model of IDEF0	6
3 IDEF Model Builder Windows	16
4 IDEF1 Data Dictionary Manager	18
5 IDEF Model Builder Status Line and Mouse Status Line	19
6 The Model Builder Interface	22
7 IDEF0 Model "Demonstration"	24
8 IDEF0: Creating an Activity Name which is too long!	25
9 IDEF0: Creating Activities	26
10 IDEF0: Creating Concepts	27
11 IDEF0: Creating Relations	29
12 IDEF0: Showing Decomposition and Context	31
13 IDEF0: Text Display Window	32
14 Model Builder with Three Panes Displayed	49
15 Symbolics Workstation Mouse	94
16 Symbolics Workstation System Status Line	95
17 Symbolics Workstation Keyboard	96

1. The first part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

2. The second part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

3. The third part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

4. The fourth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

5. The fifth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

6. The sixth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

7. The seventh part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

8. The eighth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

9. The ninth part of the document is a list of the names of the persons who have been appointed to the various positions of the Board of Directors of the Corporation.

Overview

IDEF is an abbreviation for ICAM (Integrated Computer Aided Manufacturing) Definition. There are currently three IDEF Modeling Methodologies: IDEF0, IDEF1, and IDEF2. IDEF0 is used to produce a function model which is a structured representation of the functions of a manufacturing system or environment, and of the information and objects that interrelate those functions. IDEF1 is used to produce an information model that represents the structure of information needed to support the function of a manufacturing system or environment. IDEF2 is used to produce a dynamic model which represents the time varying behavior of functions, information and resources of a manufacturing system or environment.

The Integrated Development Support Environment (IDSE) is a suite of integrated software tools that provide intelligent support for information modeling. These tools assist in function, information, and process modeling. Additional tools exist to assist in gathering and analyzing information to be modeled.

This document is a user's guide to application of the IDSE. Sections covering the requirements and design of each of the tools follow. Also, four appendices exist to describe hardware and software requirements, installation procedures, and basic hardware usage.

1. MDSE Modeling Tools Requirements and Design

1.1. The IDEF0 Modeler

Two major approaches to building IDEF0 models are in widespread use in government and industry. These approaches are referred to as top-down and bottom-up model building. A goal of the IDEF0 modeling tool is to support both of these construction techniques.

In the top-down IDEF0 modeling approach, the user has the ability to define the purpose, context, and purpose of the model and construct the A-0 diagram. The activity and ICOM flows are specified for the A0 diagram with respect to the A-0 diagram. Then the user builds the next lower level of the A0 activity by constructing each of the activities in the decomposition. When all activities that make up activity A0 are complete, the ICOM flows are added and checked for consistency with higher levels. The set of activities and flows that comprise the A0 activity are then merged into a diagram that represents the decomposition of the A0 activity. The process then continues by decomposing each the activities in the A0 decomposition (ie., A1, A2, A3, etc.) in the same manner.

The bottom-up approach to IDEF0 modeling begins with the user constructing a node tree to represent the activities of the model. The nodes can then be defined by conversion to activity diagrams. When all activities in a node diagram have been described, ICOM flows are added to create a complete activity diagram. With a subtree complete, the activity descriptions are combined into an activity diagram for the parent node. The parent nodes are then combined with other activity diagrams at the same level by linking them with ICOM flows and combining them to form the description of the next parent node. The bottom-up modeling approach then continues working upward by generalizing activities until the A-0 diagram is created.

The requirements of an IDEF0 modeling tool are to support both types of IDEF0 model construction. Facilities for creating and editing nodes in a node tree, activities in an IDEF0 model, and ICOM flows in an IDEF0 model; merging node and activity descriptions into parent nodes and activity descriptions, respectively; and decomposing nodes and activities must be provided. Additionally, models should be stored to and retrieved from persistent storage and output to hardcopy devices. These requirements are necessary for an automated tool to support IDEF0 modeling.

IDEF0 Model Development

When developing an IDEF0 model you describe activities, concepts that are related to these activities, and relations that hold between the activities. The following will describe each element of the IDEF0 model: (Refer to Figure 1)

An Activity is an action, process or operation. It is a description of "What happens in a particular environment." It can be accomplished by people, machines, or computers and usually is described by an active verb or verb phrase.

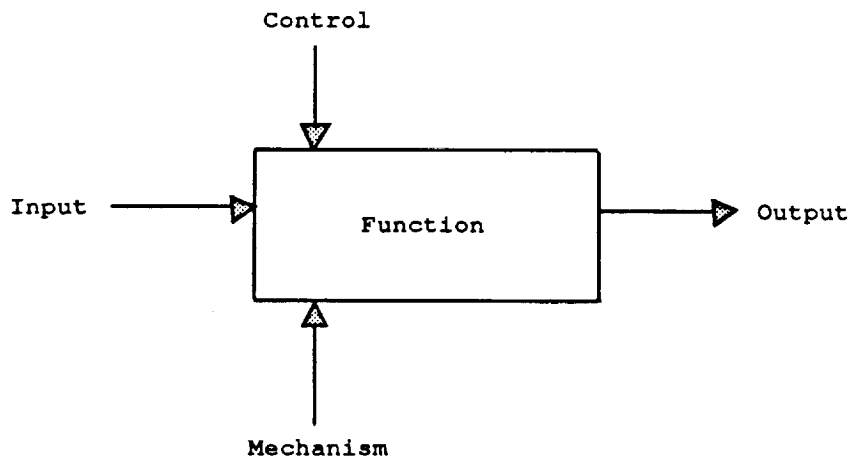


Figure 1. IDEF0 Model Development Diagram

Input to the activity represents data which undergoes a change and is transformed into an output. It is not necessary that an activity have an input. An Input is a real object or data required to perform a function. The inputs are labeled as nouns and represent physical objects, data or concepts which are transformed by a function. Inputs enter the activity box from the left hand side and the function transforms the input to an output that exits the right hand side of the activity box. The input data may be information, objects or anything that can be described with a noun phrase. One must be careful when labeling inputs, very often objects that appear to be inputs are actually controls.

Output from the activity represents data which results from or are created by a function. It is real objects that result when the function is performed. An output label is also a noun. Arrows leaving an activity box from the right show the data created when the function is performed. Output from one box can be input to another. Connecting the output of one box to the input of another box shows a data constraint. The box receiving the data is constrained since the functions cannot be performed until the data is made available via the output of the box that produced it. An output from a single box may provide some or all of the data needed by one or more functions (activities). Every activity must have an output.

Control on the activity represents data which influences or determines the transformation of inputs to outputs. A control is data that governs the accomplishment of a function. It is data which influences or determines the outputs. A control describes the conditions or circumstances that govern a function. The roles of input and control are different. The distinction between these two data flows important. A general assumption that is typically accepted by modelers is that an arrow is a control unless it ob-

viously serves on as an input. Every activity box must have at least one control arrow.

Mechanism to an activity represents a person or device which carries out a function. It is data which carries out a function. The means by which a function is performed. The mechanism arrow enters the bottom of the activity box and it is most convenient to think of the input and output illustrating what is done by a function. The control shows why it is done and the mechanism illustrates how it is to be done. Diagrams drawn without mechanisms show what functions a system must perform. Mechanisms will specify how those functions are to be performed. Mechanism arrows can be output from other activity boxes if those activities create or prepare devices as their output.

Design of IDEF0 Modeler

The design of the IDEF0 modeler is based upon the Metamodeler concept. The Metamodeler allows a description of the information to be entered and generates the basic functionality necessary for a graphics-oriented tool. An IDEF1 model of IDEF0 was developed to use as input to the Metamodeler. The IDEF0 model developed is shown in Figure 2. An appropriate description was created for the Metamodeler, supplied to the Metamodeler, and the core of the IDEF0 modeling tool was generated.

A major portion of the development of the prototype IDEF0 modeler was the user interface. The command interpreter, mouse gesture handler, and graphic display were designed and constructed to complete the IDEF0 tool. Much of the functionality was intentionally designed to run concurrently with the IDEF1 modeling tool. The user has the ability to operate on models of either modeling methodology with each loaded into the Model Builder system at the same time.

Features of IDEF0

The following list will give you an idea of some the things that you can do with the IDEF0 tool: you can create models, create an activity, move up and down a hierarchy, create concepts, create relationships, and create ICOMs to list a few. These activities will be described in further detail following this section. The IDEF0 function modeling tool:

1. Views a system as connected components, where components are functions and connections represent interfaces. This allows description of critical activities and relationships;
2. Supports the diagrams, text and glossary of IDEF0 fundamental building blocks of diagram-boxes and arrows; each box has four labelled sides: input, output, control, and mechanism;
3. Interconnects boxes and allows hierarchical representation of IDEF0 modeling;
4. Provides for cross-referencing of diagrams, text, and glossary.

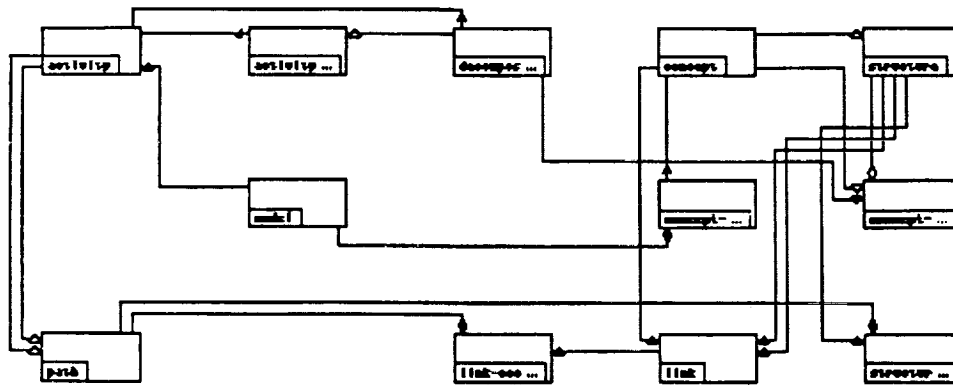


Figure 2. IDEF1 Model of IDEF0

1.2. IDEF0 Concept Editor

The Concept Editor is used not only to edit single concepts, but also to display the relationships between concepts, and where the concepts are used. Concepts are displayed alphabetically along with their subtypes and subparts.

The Concept Editor has three display panes. It displays all globally defined concepts in the left pane, the inputs (inputs, mechanisms, and controls) and outputs of the parent activity in the center pane, and the concepts used in the decomposition in the right pane.

The validity of the decomposition can be determined by comparing the parent-level concepts with the concepts used in the decomposition. All of the top-level parent concepts should have ICOM codes displayed next to them. If a parent-level concept does not have an ICOM, then that concept is either not used in the decomposition, or is not an input or output of the decomposition. In either case, the decomposition is invalid.

The inputs and outputs of the decomposition also have ICOM codes displayed with them. If an ICOM code is displayed as a double asterisk "***", then that concept is not an input or output of the parent.

Upon entering the Concept Editor, the information displayed will reflect the current decomposition displayed in the Model Builder. It is possible to move about the model while in the Concept Editor without returning to the Model Builder display. The "Change Context" command will prompt for which decomposition you wish to see. The best method of completing this choice is to use the right mouse button and selecting the decomposition from the pop-up menu. After exiting the Concept

Editor (by using the "Exit" command), the Model Builder will display the decomposition which is the current context of the Concept Editor.

Many times it is useful to see where in the model a concept is used. The "Where Used" command will list all decompositions where a concept is used. If you click the middle mouse button on a returned decomposition name, the context will change to that decomposition.

As you create concepts with types, you will note that the types of the concept inherit the parts of the parent. If a part is inherited, it will be designated with an "(i)". When parts are added or deleted from a parent, they will also be added or deleted from all children and their descendants.

It is possible for a concept to inherit a part from more than one parent. If this occurs, and the part is deleted from one parent, it will remain in the child, since it still inherits it from the other parent. If you wish to see where a concept's parts are inherited from, use "Show Inheritance". "Show Inheritance" returns a list of the concept's parts, and where they are inherited from.

1.3. IDEF0 Node Editor

The IDEF0 Node Editor is provided to allow the user to rapidly enter the skeletal structure of an IDEF0 model. The user may create activities in a new model, add activities to an existing model, rename activities, comment activities, and move activities to different locations in the model. No graphical representations of the activities are displayed and no concepts are created by the Node Editor. This allows a large portion of the model to be displayed on the screen at one time. Also, since no graphics are drawn, modifications to a model can be made quickly and efficiently.

The Node Editor displays an indented list that corresponds to the decomposition of an IDEF0 model. The further the model is decomposed, the further the list of activities is indented. For example, activity A0 appears in the leftmost margin; activities A1, A2, and A3, etc. are indented two spaces; activities A11, A12, A13, A21, A22, etc. are indented four spaces; and so on. The order in which the activities are shown follows a depth-first search pattern. For example, activities A1, A2, A3, etc. are listed beneath activity A0; activities A11, A12, A13, etc. are listed beneath activity A1; and so on. This type of display allows the user to easily see the hierarchical nature of the model being created.

The Node Editor can be entered from the Model Builder when an IDEF0 model is selected with the "Node Editor" command. The user may then create activities by supplying an activity name and number. The user can also move existing activities to different locations within the model. Those activities that existed upon entry into the Node Editor cannot be deleted -- only those that are created from within the Node Editor can be deleted from the model. This prevents existing models from being accidentally corrupted. These basic operations allow the user to very quickly create or extend a model and then switch to the Model Builder and add concepts and other information.

Another notable feature is the ability to enter comments associated with the activities. The comments are entered or modified in a small, pop-up window editor. After a comment has been created, the user can control whether or not it is displayed in the Node Editor. Thus, the user can make notes to himself about activities as they are being created.

1.4. IDEF1 Modeler

The IDEF1 modeler will extract from the Source Data List the primary adjectives in each data item. This will help establish the entity classes which will be put into the Entity Class Pool.

Using this Entity Class Pool the modeler then establishes the definitions of the Entity Classes. As a constraint on the Entity Classes, the links (Relation Classes) between the Entity Classes would be established. These Relation Classes must reflect the business rule and should allow no ambiguity of that rule. Once the Relation class has been established it would be added to the model.

Key Classes are included in the model next. The Source Data List is used to identify the candidate key class items that uniquely identify an Entity Class. This can be done by comparing the source data list and the items used to establish an Entity Class in the Entity Class Pool.

The modeler must continually check to insure that the model obeys the rules of IDEF1. All Entity Classes must have unique names. Inheritance of Attribute Classes and key classes must be insured. In the selection of the Attribute Classes in a key class the modeler will make the key class for each Entity Class unique.

A great deal of the work done by a modeler will involve the intergration of a model or a part of a model into another model. One of the major problems associated with merging or intergration of models would be the resolution of conflicts. Often there will be two model elements that are the same even though they have different names. The first step in the resolution of these conflicts is to establish a common Attribute Class pool and Glossary and a common Entity Class pool and Glossary. These are examined to determine which pairs represent the same item.

The process of merging entity classes may start with either the comparison of Entity Classes or the comparison of owned attribute classes. Choosing two model elements that are the same often includes the need to examine the textual description of the elements. The modeler may also go back to the original documentation associated with each to determine if they represent the same item. In merging two Entity Classes the modeler can compare their owned Attribute Classes. If most or all of their owned Attribute Classes are the same then the two Entity Classes most likely are the same. With the determination that two Entity Classes are the same, their relation to the surrounding Entity Classes must be resolved. Solutions must be found to any conflicts that may occur in the relation classes. The solution should preserve the business rule.

1.4.1. IDEF1 Tool Design

The IDEF1 model builder is designed to assist an IDEF1 modeler. One of the most important features of the system is the automatic enforcement of the rules of IDEF1. For instance, the key classes of an entity class must be unique, only owned attribute classes may be deleted, etc. A second key feature of the system is the ability to not only create, but also to merge models. In the merge operation, one model, or part of a model, may be copied into a second model in order to create a new model.

The model builder has been designed to maximize ease of use. Thus, for instance, command completion is supported. For most commands, only the first letter or two of the command need be typed, so that keystrokes are minimized. Also, entity class boxes may be easily selected by moving the mouse arrow across the model. The selected box will be highlighted as the mouse moves. The features of the IDEF1 Modeler are described fully in the following sections.

1.4.2. Features of the IDEF1 Tool

The required IDEF1 information modeling tool has the following features:

1. Allows modeler to focus on:
 - rules with the organization concepts
 - what people use to do their job
 - logical relationships within the organization
 - uses for problem identification
 - uses for requirements definition
2. Provides for diagram support including:
 - Entity class definition
 - Link class definition
 - Key class definition
 - Attribute class definition
 - Inherited attribute classes
 - Attribute classes in key classes
 - Inherited key classes

- Link class correspondence (strong-many-to-one, weak-many-to-one, one-to-one)
3. Allows editing of the above descriptions

1.5. IDEF1 Data Dictionary Editor

1.5.1. Project Data Dictionary

Associated with every IDEF1 model diagram is a Project Data Dictionary. A project data dictionary provides the documentation of the information sources and the audit trail for the entity and attribute classes in the model. Correct IDEF1 modeling procedure requires that a project data dictionary be developed before the model diagram is drawn. This project data dictionary is the source of all of the information that is displayed in the the model. Each project will have one Source Material Log.

A source material log is a list of all of the source material collected in the investigation phase of a project. The usual procedure in a large project is for each of the investigators to be assigned a set of source material numbers, such as 1 to 200. To each of the pieces of source material that they have they will assign one of their numbers.

In addition to the Source Material Log a project will have one Source Data List. The source data list is a list of data items that have been chosen from the information in the Source Material Log. The investigator will be assigned a set of numbers and with each item chosen as a source data item he or she will assign one of these numbers. Associated with each source data item will be a list of the source material log item numbers in which that data item is referenced.

For any project there may be multiple models, and each model will have one Entity Class Pool and one Attribute Class Pool. An Entity Class Pool is the list of potential entity classes for the model diagram. The Attribute Class pool is the list of the potential attribute classes for the model diagram. Each of these items will have a list of Source Data items with which they are associated. The reference lists for the Entity Class Pool and Attribute Class pool items along with the reference list for each of the Source Data List items provide an audit trail to the justification for the items that appear in a model diagram.

1.5.2. Project Data Dictionary Tool

The Data Dictionary System was designed to follow the procedure followed by an IDEF1 modeler. Although the user is not forced to populate a Project Data Dictionary before building a model diagram he or she must have at least created an empty one before beginning to draw the diagram. The IDEF1 diagram knows the name of the project that it is associated with and can access the correct Entity Class Pool and Attribute Class Pool as the model diagram is being designed. At any time that the modeller chooses he or she can switch to the Data Dictionary System and view the entire Project Data Dictionary.

At the time that an IDEF1 model is being built the modeler will use the list of potential entity classes as he or she populates the model with entity classes. The list of potential attribute classes is the source for most of the attribute classes that are associated with the various entity classes in the model diagram. In building a model the modeller will often think of other entity classes or attribute classes that are not in the entity class or attribute class pools. This is not unusual, but the project data dictionary will have to be updated to reflect these new items. Each item listed in a project data dictionary must have a description or definition. This definition in the case of the entity class pool and attribute class pool items must be the same as that in the model diagram. The commands in the project data dictionary have been designed to respond to these requirements.

2. MDSE Model Development Support Requirements and Design

The following is a detailed description of the features of the Model Builder:

Command Completion: All commands that are available in the system have a command completion feature. The user needs to type only the first letter or two of the words in a command for the system to complete the command. Included in this feature is the use of the mouse to select user completions of commands. For the user's convenience, only those completions that are acceptable within the methodology will be mouse sensitive. The system will prompt the user to select the appropriate completion. For instance, if the user is attempting to add Attributes to a Key Class under IDEF1, only Key Classes will be mouse sensitive.

Structured English Input: All commands to the system clearly state the operation that will be performed given the correct user completion. They are very English like. For instance if the user wishes to add a new entity class to the model the command is "Create an Entity Class [an entity class name]." The user completes this command with the name of the entity class.

Free Form Layout: Under IDEF1 the user may place the entity class boxes at the point of his/her choice. This operation is done by clicking with the mouse at the position in the model that the user wishes to position the box. Under IDEF0 the activities are placed automatically due to IDEF0's strict formatting rules.

Auto Router: The Auto Router feature of the tool uses its knowledge of the model to automatically route between activities or entity classes in such a way that the links are placed in an optimal fashion.

Auto Layout: The auto layout feature of the tool positions IDEF1 entity class boxes in order of their dependence. The dependent entity classes are automatically positioned below independent entity classes. IDEF0 activities are automatically placed along the diagonal.

Model Object Save / Hardcopy: Complete or partially complete models may be saved for future editing and review. Restoration of models is accomplished by loading the previously saved file. The system includes a hard-copy feature of both the graphical model and the associated kit.

Interactive Graphics (Display & Edit): All model elements that are acceptable completions for a command are selectable with the mouse. A middle mouse click on any entity class will display the unique name of that entity class, its key class, and all of its attribute classes. Clicking middle on a concept name shows the complete name along with the subtypes and subparts of that activity.

File Output of Graphics: Saving a model includes the saving of the graphical description of the model. When a file is restored the graphics is restored in the same configuration that it was during the latest editing session.

On Line Help: The <HELP> key on the keyboard provides mousable access to all commands and acceptable command completions. Because of the use of structured English in the design of the system commands, only a knowledge of IDEF1 or IDEF0 is required to use the system. After using the <HELP> key to see the available system command the use of the middle mouse button provides additional help on an individual command. This mousability of commands viewed via the <HELP> key allows first time users to start producing valid models as they learn the tool.

2.1. Model Validation and Support

Automatic Model Rule Enforcement: One of the major features of the Model Builder is automatic model rule enforcement. A user is not allowed to enter commands or command completions that will violate IDEF methodology rules. For instance the user must assign unique names to each entity class when developing a an IDEF1 model.

Guaranteed Model Consistency: The Model Builder guarantees that the model will remain consistent with the methodology in the event that model elements are moved and deleted. The removal of an activity from the model will remove all the activities children. All attribute classes that are inherited by dependent entity classes will be automatically removed. The removal of a relation class from a model will automatically remove any attribute classes that are inherited along that link.

IDEF1 English Language Statements of Model Assertions: English language statements are automatically created for all model assertions for the model reviewer. English language statements are printed on a form for the reviewer to agree or disagree with.

IDEF0 Decomposition Validation: When completed with a decomposition, the "Check Decomposition" command will notify the user of any design inconsistencies.

ZMACS Editor for entering Descriptions and Definitions: A ZMACS editor is provided for entering model element descriptions and definitions. These Descriptions and definitions are then included in the kit. **Graphical Browsing / Editing:** The tool has graphical scrolling. By completion of a single command the screen will scroll to the position in the model where a particular model element is located. The modeler may edit the appearance of the model diagram by moving any entity class to another position within the diagram or changing the entry/exit points for relation class lines. The entire model diagram can be reformatted by using the automatic formatting feature of the tool.

Kit Production: One command will generate an IDEF1 kit consisting of attribute class diagrams, attribute class definitions, entity class definitions, etc. The command "Generate Report" will generate the entire kit for the selected model.

Wall Chart Production: A model diagram can be printed by completing the command "Hardcopy Display." The system prompts the user for the "zoom factor" for the printout. This will produce the entire model diagram on one or more 8 1/2 x 11 pages.

Commenting Support: As a reviewer reviews a model there is a need to make comments on the model. The Model Builder brings up automatically a ZMACS editor for these comments. The comments are stored with the model and attached to each model element, model, or model view that has a comment.

2.2. Model Integration Support

One of the major features of the Model Builder is integration of multiple models. The Model Builder provides support in all activities involved in the integration of multiple models.

Model Copy: All or part of one model may be copied into another model. IDEF1 model copy features a choice of spaghetti pull or a snip-it feature. The spaghetti pull will bring all of the dependent entity classes along with it. If the snip-it choice is made, the inherited attribute classes are collapsed into the entity class that were selected for copying. IDEF0 Model copy has been implemented as the Copy Model Portion command.

Model Merge: The IDEF1 tool allows the copying of two models into a single model. When the copy model command is executed the constraints imposed normally by the tool are relaxed so that the system may contain for a brief period of time some inconsistencies. After the copy is completed individual entity classes may be merged by executing the command Merge Entity Classes. The model merge command has not yet been implemented for IDEF0 models.

IDEF1 Model Element Merge: The modeler using the tool will indicate to the system which of the following elements to merge.

1. **Entity Class:** Two entity classes may be merged. The system will request which is to remain. This is the one whose name and label will be in the model being developed.
2. **Attribute Class:** Two attribute classes may be merged in a manner similar to that for entity classes. The system will use the name that the user specifies. The system will generate statistical information on the attribute classes to assist the modeler in determining which

Attribute classes should be merged. This command has not yet been fully implemented.

3. **Relation Class:** All of the original relation classes are retained when two entity classes are merged. Those Relation classes between two merged entity classes may be explicitly deleted or modified by the modeler. This command Merge Relation Classes has not yet been implemented.

2.3. Model Application / Analysis Support

The following describes support provided by the current version of the IDEF1 modeler in the areas of "model application" and "model analysis". Model application refers to the use of a model after it has been constructed. The current meta modeler component of the Model Builder supports the automatic generation of acquisition and management systems from a model. Model analysis support refers to support provided for identification of common data between two models, or identification of duplicate information within a model.

2.4. Architectural Issues

The following section describes some of the key construction features of the current Model Builder:

Presentation Types: Each of the model elements has a presentation type. Because presentation types are used with the dynamic windows the semantic content of the objects displayed in the windows is remembered by the system. This feature (presentation types and dynamic windows) means that the displayed presentations are automatically mouse sensitive when their type is appropriate as input to the system.

Dynamic Windows: The dynamic window feature of the Symbolics provides for scrolling within the window in both a horizontal and vertical directions. This means that an entire model is loaded into a window even if only a portion of it may be viewed at one time.

2.5. IDEF Model Builder Windows

The IDEF Model Builder has three windows (see Figure 3

- Model Builder Development Window.
- Command Window.

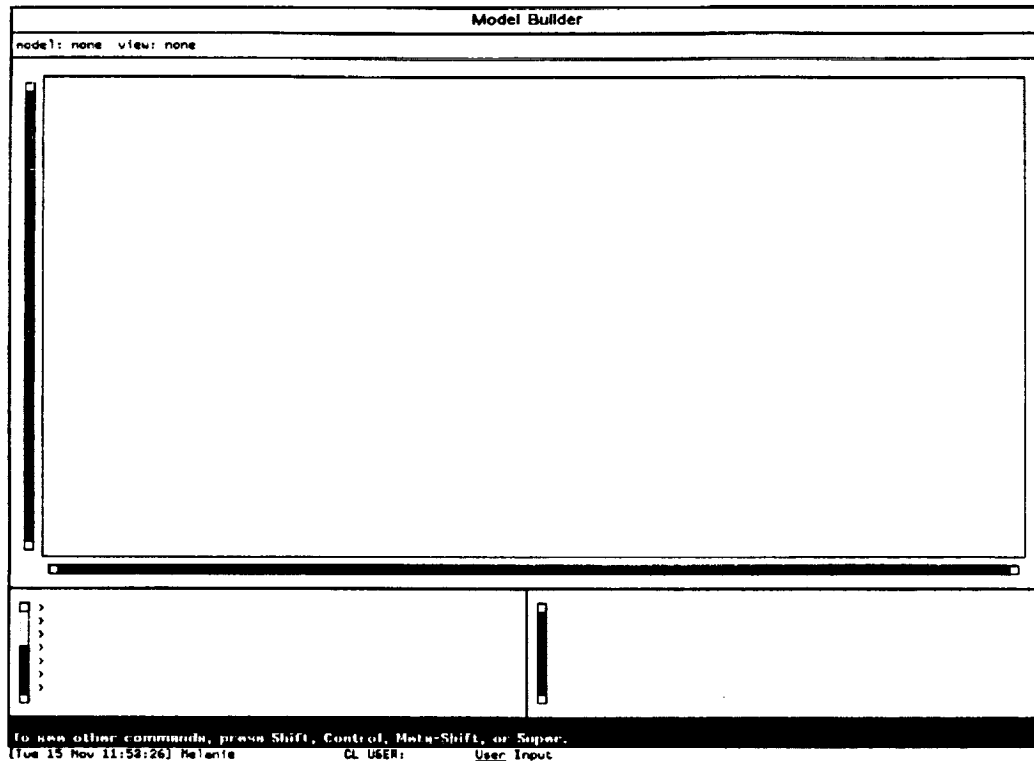


Figure 3. IDEF Model Builder Windows

- Text Display Window.

The descriptions of these windows are as follows:

Model Builder Development Window: This is the main window of this development tool. Here is where you will create all of your models, be it IDEF0 or IDEF1. This pane can be split into as many as three panes to accommodate several models at any one time. You will notice that there are scroll bars that run across the bottom and the side of the pane. These are used to help position the model at any point in the pane that you wish to view. It might be necessary for you to take advantage of this scrolling feature periodically because the IDEF Modeling Tool does not have the capability to display a complete model in the model builder development window. In order for you to see your complete model at one time you will have to get a hardcopy.

Command Window: This is the window in which you will issue all of your commands. Help will also be displayed here along with any error messages. This window also has a scroll bar on the left side which you can operate one of two ways: 1). you can move the mouse arrow into that bar and it will turn into an up/down arrow which can be controlled by mousing right to move up or mousing left to move down, (you can get further help by looking at the mouse status line for scrolling control) 2). or you can control scrolling by using the META-SCROLL keys for moving up and SCROLL for moving down.

Text Display Window: This window is used to display all messages that commands may produce except error messages. This window also has a scroll bar on the left side which operates in the same manner as the command window scroll bar described above.

The Data Dictionary Manager has four main windows (refer to Figure 4):

- Source Material Log
- Source Data List
- Entity Class Pool
- Attribute Class Pool

These windows are where you will store the Data Dictionary for a project. For example, the entity class and attribute class information created here can be used for IDEF1 modeling as long as the model names are the same.

The Data Dictionary Manager contains a list of projects, which at any one time will contain a current project and current model (meaning the one that is currently being displayed). This current project will have; a name, a source material log, a source data list, and a list of models. These models will have; a name, an entity class pool, and an attribute class pool.

2.6. The Temporary Windows

ZMACS Editing Window: This window is used in the IDEF0 model diagrammer, IDEF1 model diagrammer, and the Data Dictionary Manager. It is a temporary window that the AUTOIDEF system uses for the insertion of all element descriptions and comments. It is a full ZMACS editor.

View Only Window: This window is used in the Data Dictionary Manager to view all of the information that the system has on any of the items that are visible on the screen. No editing is possible. Hold down the META key and move the mouse arrow over the item to be viewed and middle click with the mouse and all of the details of the chosen item will be displayed in this window.

Source Material Log Project Data Dictionary: NONE		Source Data List Project Data Dictionary: NONE		Entity Class Pool for Model: None Defined		Attribute Class Pool for Model: None Defined	
Number	Item Name	Number	Item Name	Number	Item Name	Number	Item Name

Type any character to continue:
>IDEF1 Data Dictionary Editor
>

Mouse-R: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
(Wed 16 Nov 8:33:31) Melanie CL USER: User Input

Figure 4. IDEF1 Data Dictionary Manager

2.7. IDEF Model Builder Status Line and Mouse Status Line

Two other features of the model builder interface are (see figure 5):

- Model Builder Status Line
- Mouse Status Line

Model Builder Status Line: Gives the user two pieces of information:

1. **Model:** Will have beside it the name that you gave to the model, and
2. **View:** Will tell you where in the decomposition of the model you are.

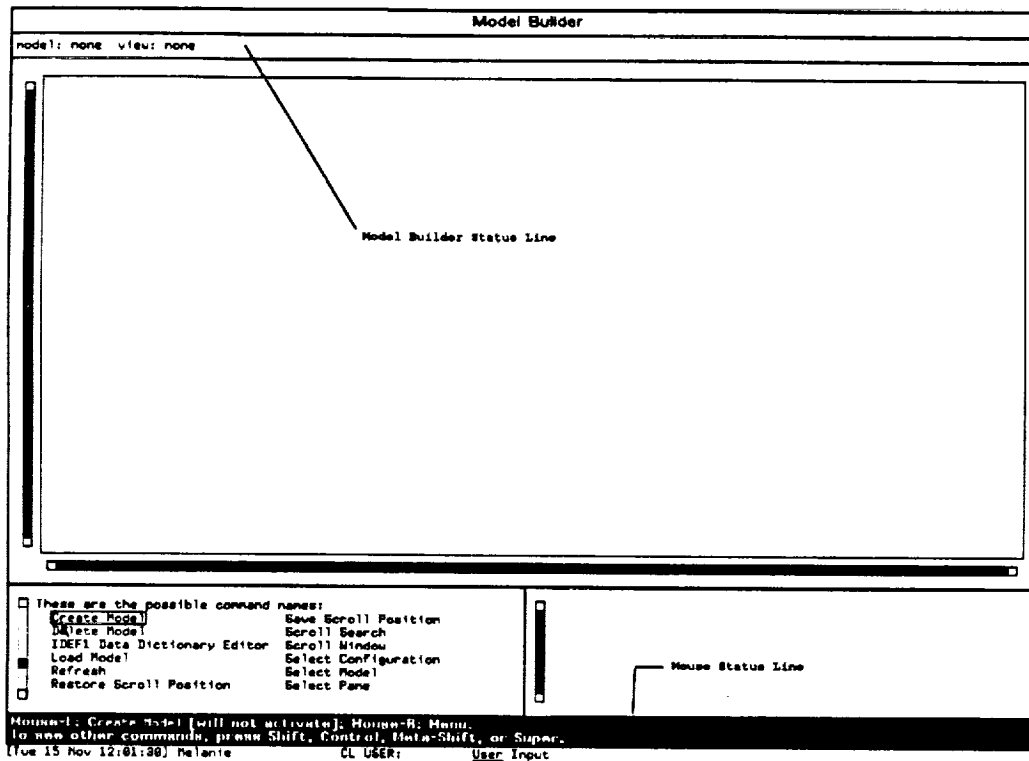


Figure 5. IDEF Model Builder Status Line and Mouse Status Line

Mouse Status Line: Will give you information on the mouse buttons (L (Left), R (Right), and M (Middle)). For example, if you create an activity in IDEF0 and you place the mouse arrow over the activity box, look at the mouse status line, You will see something like: Mouse-L: Show Decomposition (click on an activity) A1:"first"; Mouse-R: Menu. This message is saying that if you click left on the activity box A1 (named "first"), you will get its decomposition. By pressing the SHIFT key you can see what double clicks will do.

ORIGINAL PAGE IS
OF POOR QUALITY

3. Getting Started on the MDSE

Before you can begin your session, you must make sure that the machine is up and running correctly. If you have any difficulty with this refer to the instructions that begin on page 97 . It is probably best to re-boot the machine when you first start (see Cold Booting page 99). Unless you know what the previous user was doing, there may be conflicting programs loaded in the system. If the previous user was using the IDEF Modeling Tool, you won't need to re-boot. You must also be logged in order to access the file system. This puts your user name on print-outs and files you create and lets other users know that you are using the machine.

3.1. Running IDEF0 and IDEF1

To initiate execution follow the procedure below:

1. Login.
2. At the command prompt, if the model builder is not already loaded, you will need to load in the files of the IDEF modeling tool before you can access it. Type the following command:

- Load System Model-BUILDER

You will see:

Load System (a system) Model-BUILDER

If you are the first user since the machine was halted (cold-booted) you will need to load in the system files. But if someone before you has used the IDEF Model Builder, then it will not be necessary to re-load the system files. It will take approximately 25 minutes to load all the system files.

3. While the files are being loaded, you will be prompted by two questions which are warnings by the system about system functions that are trying to be changed or redefined. The first warning you will see:

Warning: Function (FLAVOR:METHOD:ADD-GRAPHICS- PRESENTATION DYNAMIC-WINDOW) being redefined by file MODEL-BUILDER:MODEL-BUILDER;IDEF1; DW-MODS, was previously defined by file SYS:DYNAMIC-WINDOWS;DYNAMIC-WINDOW. OK? (Y, P, or N)

If you would like help on what the different selections will initiate, press the <HELP> key and you will get a list of the choices with their definitions.

4. Choose **Y** to proceed to redefine, **N** to not redefine it, and **P** to proceed and not ask in the future (for this pair of files). You should select **P** but the system will not bomb if you choose **Y** instead. Do not choose **N**, the system functions are being redefined for the IDEF model builder and things may not work exactly like they are supposed to if you respond with **N**.
5. The second warning that you will see:

Warning: Function COMPLETE-STRING-BOUNDS, being redefined by file MODEL-BUILDER:MODEL-BUILDER;ZWEI-HELP-PATCH, was previously defined by file SYS:ZWEI;MINI-BUFFER. OK? (Y, P or N)

Here again you should select **P**. These are the only two warnings that you will encounter while the system files are being loaded. If **P** is pressed several times when loading beings, the user need not watch the loading process.

6. Once the files have been loaded, press **<SELECT>-3** to bring up the Model Builder. It will take approximately 20 to 30 seconds to load, and the initial screen should appear. Refer to Figure 6 .

This screen will look the same for both IDEF0 and IDEF1. The only difference is the commands that each tool uses.

7. If, for any reason, you want to interrupt execution before the system reaches a normal stopping point, you can hit the **<SUSPEND>** key on the top right hand part of the keyboard, and the system will stop as soon as the current function has completed its actions. Execution can be resumed from that point by pressing **<RESUME>**.

Logging in: Logging in is an initial procedure that is necessary in order to run IDEF1. The following steps should be followed when logging in:

1. Type **Login <SPACE BAR>** -- You will see: Login (user name).
2. Type in your Login name.
3. Press **<RETURN>**

You will see: Loading [machine ID]:[Your Login Name] lisp-init.lisp into package USER (really COMMON-LISP-USER).''

Getting the Complete Pathname: If you are planning on editing or reviewing a previously saved file you will need to enter the complete pathname for the file that contains the model. The complete pathname is also needed when you save a model. The Load and Save commands both require the

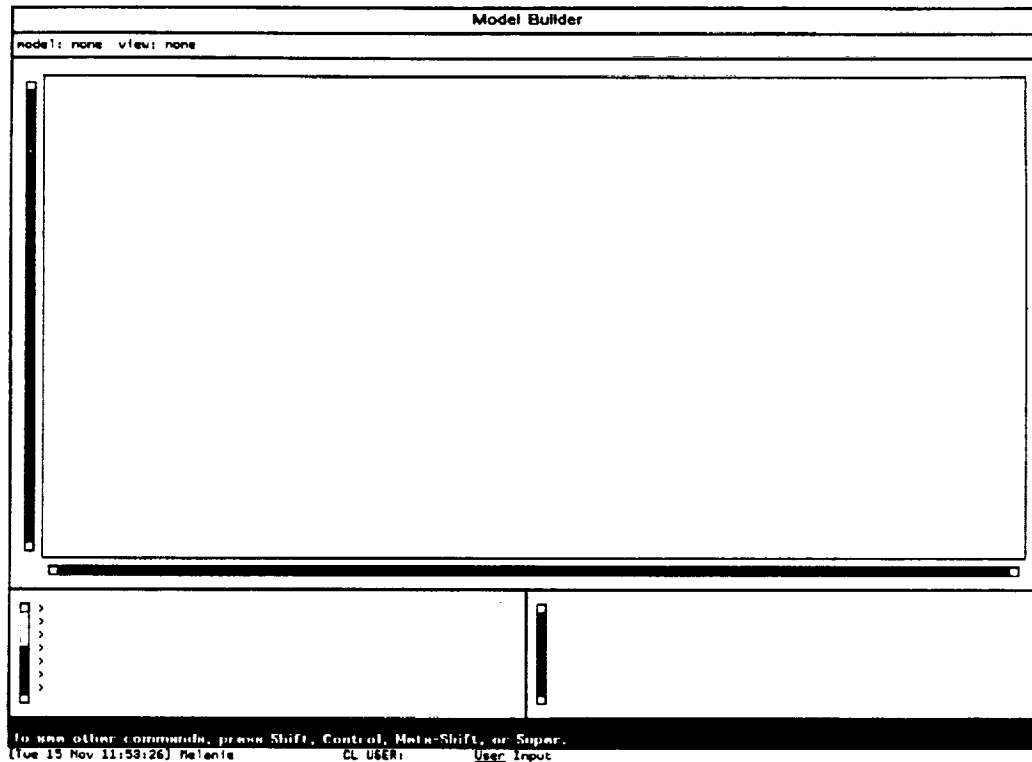


Figure 6. The Model Builder Interface

complete pathname for the file. The format for a complete pathname is: [machine name]: [user directory] [filename.lisp]. If the file is in a sub-directory the complete pathname would be: [machine name]: [user directory] [sub-directory] [filename.lisp].

The File System: If you want to look at the files in your directory, or print out one of them, Type <SELECT>-F. This will bring up the file system window (without disturbing work you are doing in another process, such as the Model Builder. Select the "Root Directory" option from the menu at the top of the screen. This will display a list of all top level directories. To open your directory, click left with the mouse over the directory name, and a list of your files will be displayed. If you wish to print a file, click the right mouse button with the arrow placed over the file name. This will display a menu of actions that can be taken on that file. To get a printout, just select the "Hardcopy" option with a left mouse click over the item.

ORIGINAL PAGE IS
OF POOR QUALITY

Ending a Session: When you have finished a session all modified models should be saved and you should logoff the computer. The steps to follow are:

1. Save all models that have been modified using the command **Save Model** (see page 48 for details).
2. Type **<SELECT>-L**. This takes you to the Lisp Listener.
3. Type **Logout <RETURN>**.

3.2. Sample Usage Scenario for IDEF0

This section provides you with step-by-step instructions to perform various tasks using the IDEF0 Model Builder. Items that are shown in **boldface** are to be typed in. The items in parentheses are prompts that you will be given when you press the **<SPACE BAR>**.

3.2.1. Creating a Model, Activities, Concepts, and Relations

In this example you will create an IDEF0 model, create activities, create concepts, and create relations. Follow the steps below to create this sample business model called "Demonstration."

1. Once you are in the IDEF Model Builder (**<Select>-3**), you are ready to create a model. In the "Command Window" type:
 - **Create Model (Idef0 or Idef1) IDEF0 (name of non-existent model) demonstration**

The system has a command completion feature that allows you to type the first few letters of a command and then, by pressing the **<SPACE BAR>**, it will complete it for you as long as it recognizes the command. If it does not complete the command, make sure you have typed the partial command correctly.

The name of the model (demonstration) will now appear in the model builder status line along with the view (A-0). (Refer to Figure 7) Now you are ready to create your activities.

2. To create an activity, type the following:
 - **Create Activity (name of non-existent activity) Manufacturing (activity number (e.g. A0))A1 <RETURN>**

NOTE: The commands are not case sensitive and remember that you have command completion.

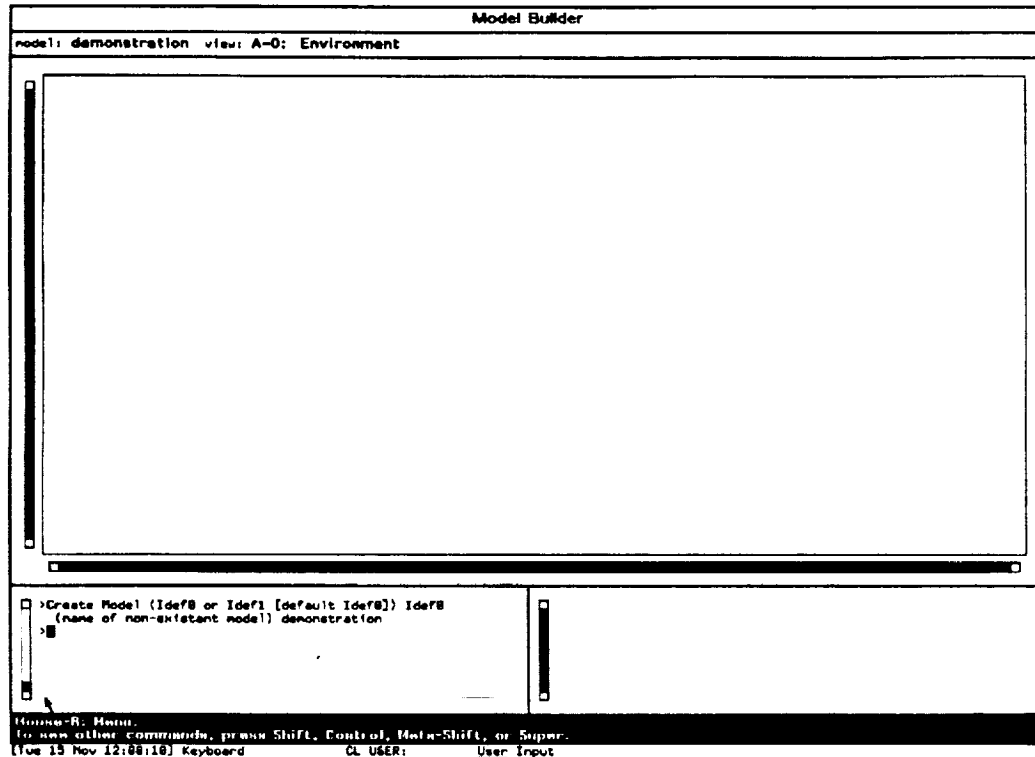


Figure 7. IDEF0 Model "Demonstration"

Because manufacturing is a long word, and the activity boxes cannot handle words with eleven or more characters, you will be prompted with a message asking you to hyphenate manufacturing or abbreviate it and end it with a period. (Refer to Figure 8) You will see:

"MANUFACTURING" is too long.
 Enter hyphenated (end with -) or
 abbreviated (end with .) form of the word using
 less than 11 characters:

3.

Now, if you want to hyphenate the word, type:

ORIGINAL PAGE IS
OF POOR QUALITY

- **manufac- <RETURN>**

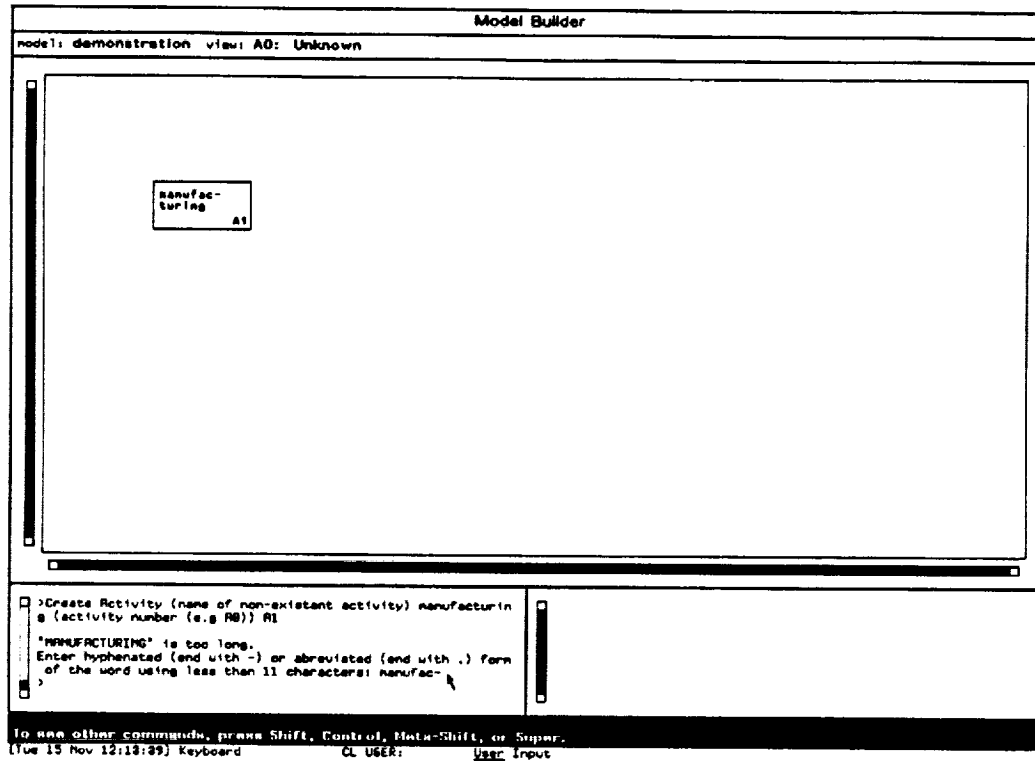


Figure 8. IDEF0: Creating an Activity Name which is too long!

4. Now we can create the rest of the activity boxes: (Refer to Figure 9)
5.
 - Create Activity marketing A2
 - Create Activity sales A3
6. Before you can create any relations, you must first create the concepts. Every concept has 2 subcategories; types and parts. You can have a concept without any "types" or "parts," with only "types" and no "parts," or with only "parts" and no "types." Let's create our concepts: (Refer to Figure 10)

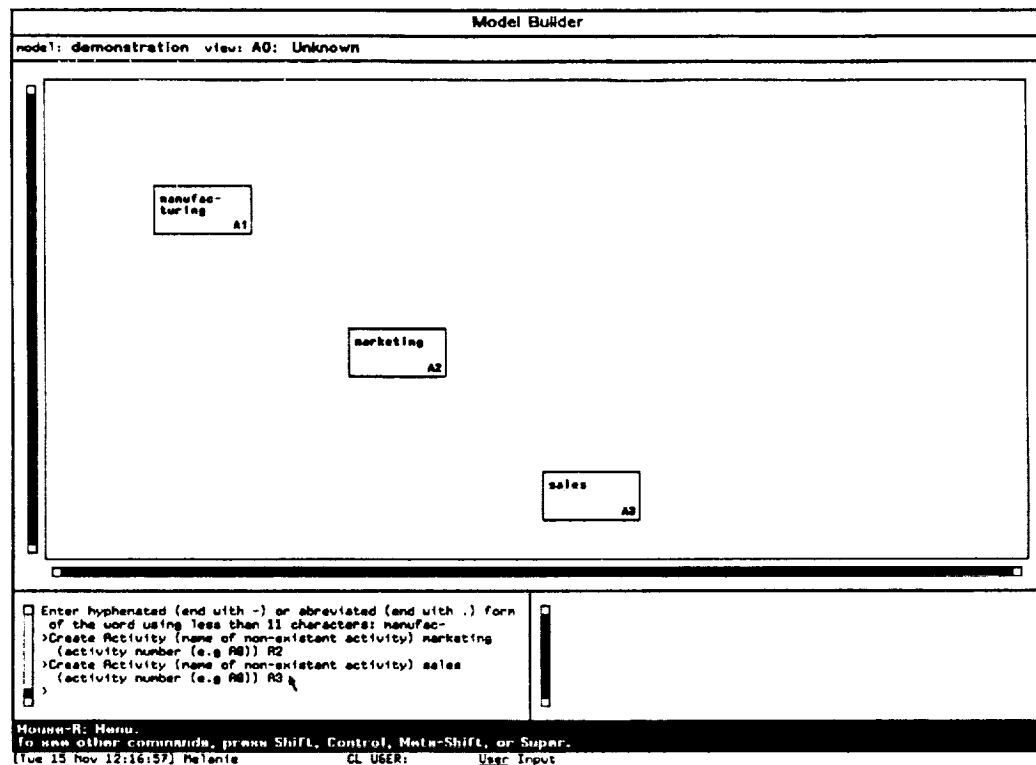


Figure 9. IDEF0: Creating Activities

NOTE: All "parts" and "types" are also created as concepts. Several concepts can be created in a Create Concept command. They are listed with a comma between each concept. Remember that this will include parts and types.

- Create Concept product, value, quantity, computers, software
- Create Concept surveys, advertising, pamphlets

Concepts can be created in the Concept Editor as well as in the top level of the Model Builder. Enter the Concept Editor by typing

- Concept Editor <RETURN>

Now, use the same method to create the concepts: money, \$, capital, revenue, raw materials

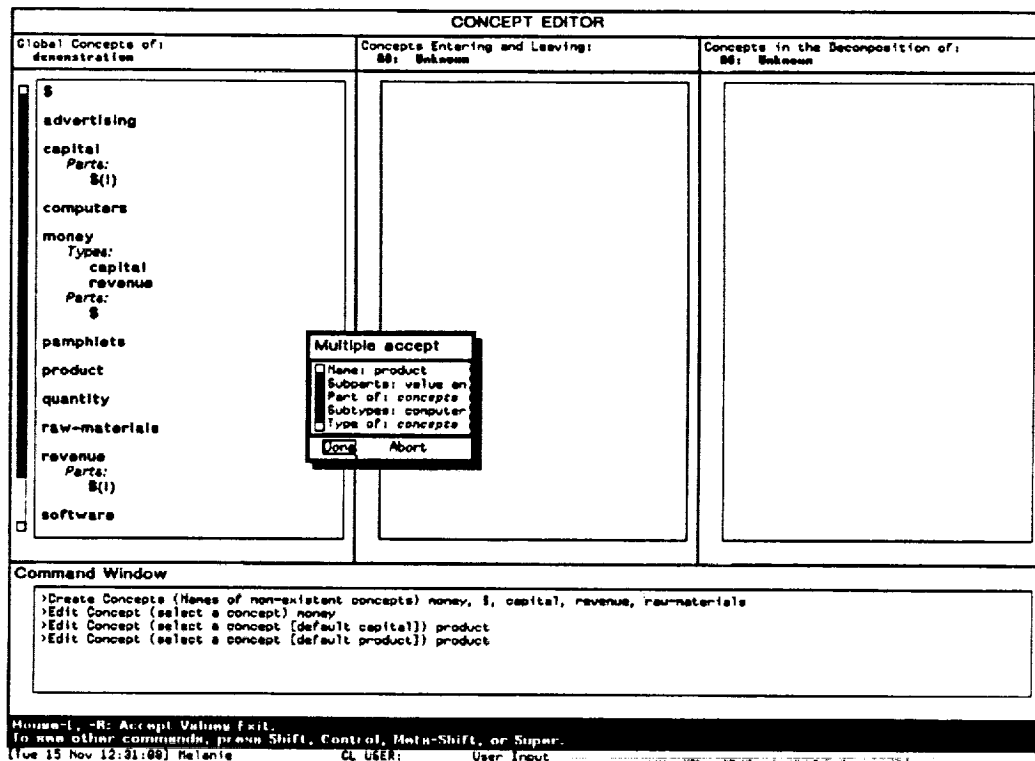


Figure 10. IDEF0: Creating Concepts

7. Within the Concept Editor, you can assign parts and types to concepts. This is done with the Edit Concept command either in the Editor or in the top level of the Model Builder. Type:

- **Edit Concept Money**

You will see:

Edit Concept (Select a concept) Money

The Multiple Accept box will appear on the screen (See Figure 10). Use the mouse to select: *Subparts: concept* and type:

- \$

Then select: *Subtypes: concept* and type:

- revenue, capital

Select *DONE*. Now, use the same method to edit the concept, "product." The subparts are "value" and "quantity" and the subtypes are "computers" and "software." Next, leave the Concept Editor by typing:

- Exit Concept Editor <RETURN>.

NOTE: There are a few important things that should be mentioned about activities. "A0" is the top level activity and if you should try to delete this activity, you will delete your whole model (beware). This will only delete you activities, your concepts will still be defined. With IDEF0 every decomposition must have at least 3 activities and not more than 6. Six is the maximum number of activities IDEF0 will allow you to have on one level. Also, it is important that every activity have a control and an output; inputs and mechanisms are optional (these will be discussed in further detail later).

8. Now we can start to create our relations. Type the following commands: (Refer to Figure 11)

9. NOTE: You type the **boldface** followed by <SPACE BAR>, and the prompts will appear. Each of these sets is ended with <RETURN>.

- Create Relation A1 (produces) **product** (as [default Control]) **Output** (activity) **A-0**
- Create Relation A1 (produces) **value** (as [default Control]) **Control** (activity) **A2**
- Create Relation A1 (produces) **quantity** (as [default Control]) **Control** (activity) **A3**
- Create Relation A-0 (produces) **surveys** (as [default Control]) **Input** (activity) **A2**
- Create Relation A2 (produces) **advertising** (as [default Control]) **Output** (activity) **A-0**
- Create Relation A2 (produces) **pamphlets** (as [default Control]) **Input** (activity) **A3**

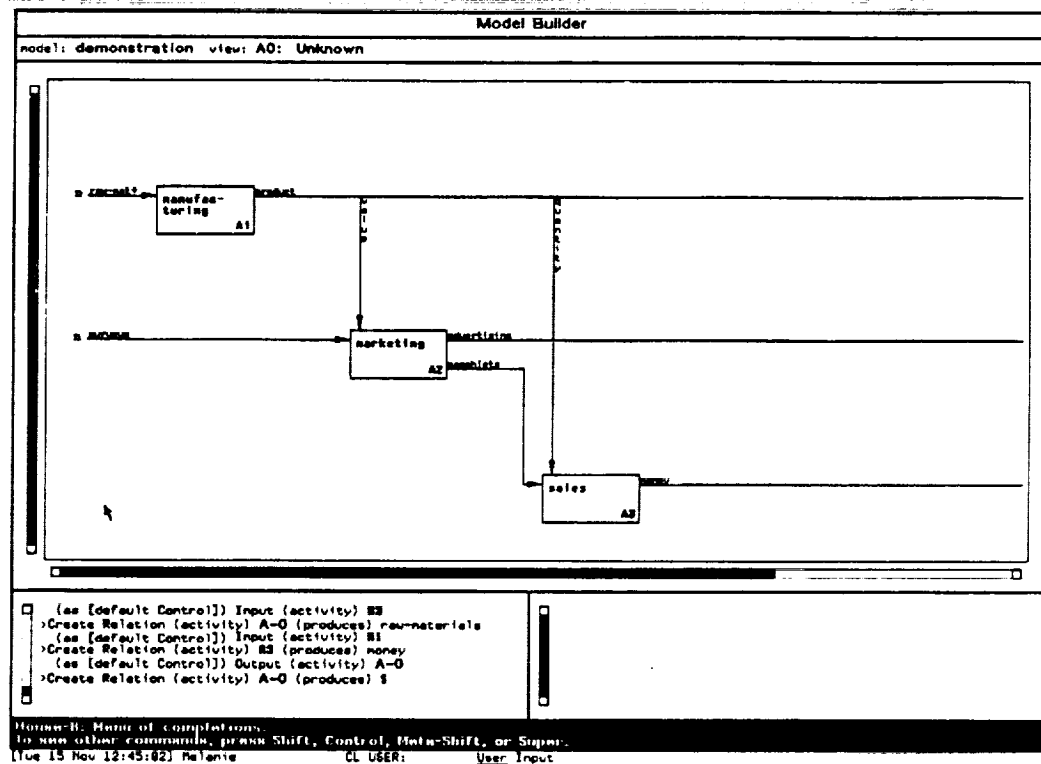


Figure 11. IDEF0: Creating Relations

- **Create Relation A-0 (produces) raw materials** (as [default Control]) **Input (activity) A1**
 - **Create Relation A3 (produces) money** (as [default Control]) **Output (activity) A-0**
 - **Create Relation A-0 (produces) \$** (as [default Control]) **Control (activity) A1**
10. The demonstration model should be complete. One thing that we should mention now is if you would like to see the decomposition of your sample model, you can do this by using the commands,
- **Show Decomposition, and**

- **Show Context**

Show Decomposition will move you down in the heirarchy and Show Context will move you up in the heirarchy. All you need to do is place the mouse arrow over any activity box; for this demo, the manufacturing box. Click left with the mouse (Show Decomposition) and it will take you down a level. Because activity A1 does not have any lower level decomposition, you will get a blank screen. To move back to the level that you came from, type **Show Context**. You can move up and down these levels with the mouse or by typing in the commands in the command window. A double click left will move you up a level (Show Context) but only if there is an activity box to click on, otherwise you must type in the command in the "Command Window". Refer to Figure 12 .

11. If you place the mouse arrow over a label on a relation, you can have the concept associated with the label described by its parts and types. Place the mouse arrow over *product*, and a box will appear around it. Click Mouse-M once and look in the Text Display Window. Product will be listed with all of its types and parts. Refer to Figure 13 .
12. You should save your model periodically throughout your session so you will not lose all of your work should something go wrong. To save your model type:
 - **Save Model** (Name of existing model [default demonstration]) **demonstration** (to pathname [default [machine name]: [login name] foo.lisp]) [**machine name**]: [**directory name**] **demonstration.lisp** <RETURN>
13. To leave the IDEF Model Builder all you have to do is press <SELECT>-L. This will return you to the Lisp Listener.

3.2.2. Hints and Assumptions of IDEF0

The following are points to remember when creating your IDEF0 models:

1. When creating your activities, every decomposition must have at least three activities but no more than six. Except the decomposition of A-0 which only contains one activity A0.
2. You should be very careful when deleting an activity, everything contained in that portion of the tree structure. Never delete activity A0, even if you do not wish to name it (you can leave it as an unknown). If you delete it you will delete your whole model (everything that you have created in lower levels will be gone).
3. With some commands it is possible to click right and have a "Select Completion" menu come up so that you can make your selections from it without typing in the command. In the case of creating relations, sometimes you may

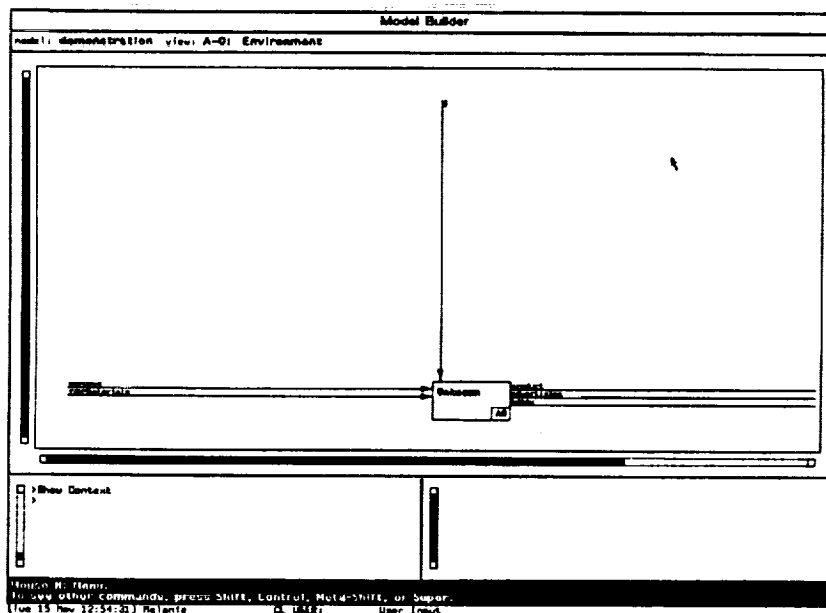
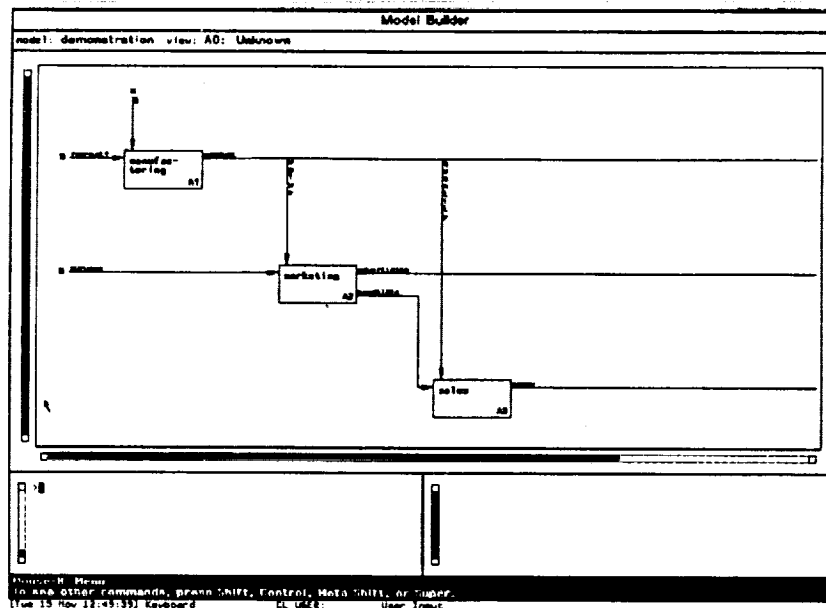


Figure 12. IDEF0: Showing Decomposition and Context

not remember all the concepts that you created previously. So when you get to the part where it ask for what the activity produces, you can click right and get this "Select Completion" menu of all the possible concepts.

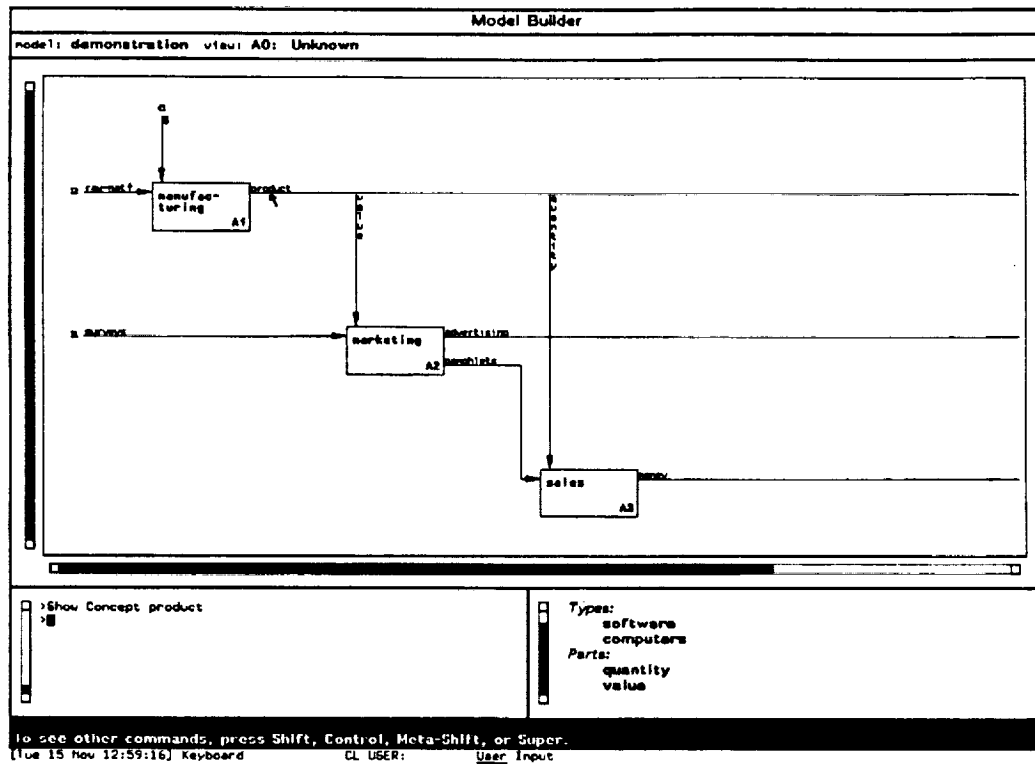


Figure 13. IDEF0: Text Display Window

4. Remember that you have command completion so that you do not have to type out all your commands. Simply type in the first few letters and then press the <SPACE BAR>.

3.3. Sample Scenario for IDEF1

This section provides step-by-step instructions for the creation of an IDEF1 model called Purchasing for a project called ABC Manufacturing.

1. First, a Project Data Dictionary must be created. From within the Model Builder, type:
 - IDEF1 Data Dictionary Editor <RETURN>

- **Create New Project Data Dictionary** (Name of New Project Data Dictionary [Default ' ']) **ABC Manufacturing** <RETURN>
- **Create New Model Current Project Data Dictionary** (Name of New Model) - **Purchasing**

2. Now, we will create Source Material Log Items. To do this, type:

- **Create Source Material Log Item** <RETURN>

A menu entitled *Declare Source Material Log Item Parameters* will appear.

- Select *Source Material Number* using the mouse and type: **1**
- Select *Source Item Name* and type: **"Order Form"**
- Select *Contributor's Name* and type your name.
- Select *Done*

The editor will appear. Type in a definition and description of the document, such as,

- **Blank form. Used to place orders for items purchased.**

Press <END> and the editor will save the description and disappear. Now, use the same process to enter the following Source Material Log Items:

Source Material Number: **2**
 Source Item Name: **Bill of Materials**
 Contributor's Name: Your Name
 Description: **Product description by parts.**

Source Material Number: **3**
 Source Item Name: **Inventory Record**
 Contributor's Name: Your Name
 Description:

Source Material Number: **4**
 Source Item Name: **Vendor List**
 Contributor's Name: Your Name
 Description: **List of companies from which various parts may be purchased.**

Source Material Number: **5**
 Source Item Name: **Employee Record**
 Contributor's Name: Your Name
 Description:

3. Next, Source Data Items will be created. Type the following:

- **Create Source Data Item** (A Source Data Item Label) **"Name of Part"**
(Source Data Number) **1** (Name of Person Creating this Source Data Item)
"Your Name" (One or More Source Material Reference Numbers)

Choose the Reference Numbers (2) and (1) from the Source Material Log using the Mouse.

IMPORTANT: The Source Material Reference Numbers must be selected with the Mouse from the Source Material Log.

When you are finished, press <RETURN>. The editor will appear, and you should type a description and definition of the item. Press <END>, and the editor will disappear. Now, in the same manner, please enter the following Source Data Items:

Source Data Number: **2**
Source Data Item Label: **"Part Number"**
Contributor's Name: **"Your Name"**
Reference Numbers: **2 1**
Description: **Inventory number of the part.**

Source Data Number: **3**
Source Data Item Label: **"Quantity of Part in Stock"**
Contributor's Name: **"Your Name"**
Reference Numbers: **2**
Description:

Source Data Number: **4**
Source Data Item Label: **Quantity**
Contributor's Name: **"Your Name"**
Reference Numbers: **1**
Description:

Source Data Number: **5**
Source Data Item Label: **Price**
Contributor's Name: **"Your Name"**
Reference Numbers: **1 3**
Description:

Source Data Number: **6**
Source Data Item Label: **"Part Description"**
Contributor's Name: **"Your Name"**

Reference Numbers: 1 3

Description:

Source Data Number: 7

Source Data Item Label: "Authorization Code"

Contributor's Name: "Your Name"

Reference Numbers: 1

Description:

Source Data Number: 8

Source Data Item Label: "Purchase Order Number"

Contributor's Name: Your Name

Reference Numbers: 1

Description:

4. Next, we will create an Entity Class Pool. To do this, type the following:

- **Create Item for Entity Class Pool** (An Entity Class Name) **Part** (An Entity Class Number) **1** (Name of Person Creating this Entity Class Item) **Your Name** (One or More Source Material Reference Numbers)

As before, the Reference Numbers must be chosen from the Source Material Log using the Mouse. Please select Reference Numbers 1 6 2 3 and then press RETURN. When the editor appears, type in a definition and description of the item.

Now, please follow the same procedure for the following Entity Class Items:

NOTE: Item and Contributor's Names must be entered with no spaces (such as Your-Name) unless they are enclosed in quotes (such as "Your Name").

- 5.

Entity Class Number: 2

Entity Class Name: "Purchase Order"

Contributor: "Your Name"

Source Data Reference Numbers: 8 7 1

Description:

Entity Class Number: 3

Entity Class Name: Item

Contributor: "Your Name"

Source Data Reference Numbers: 4 5

Description:

Entity Class Number: 4
Entity Class Name: **Department**
Contributor: "Your Name"
Source Data Reference Numbers: 7
Description:

Entity Class Number: 5
Entity Class Name: **Buyer**
Contributor: "Your Name"
Source Data Reference Numbers: 7
Description:

Entity Class Number: 6
Entity Class Name: **Employee**
Contributor: "Your Name"
Source Data Reference Numbers: 12
Description:

Entity Class Number: 7
Entity Class Name: **"Purchase Order Item"**
Contributor: "Your Name"
Source Data Reference Numbers: 6 5
Description:

6. Next, we will create an Attribute Class Pool. To do this, type the following:

- **Create Item for Attribute Class Pool** (An Attribute Class Name) **Buyer**
(An Attribute Class Number) 1 (Name of Person Creating this Attribute
Class Item) **"Your Name"** (One or More Source Data Reference Numbers)

As before, the Reference Numbers must be chosen from the Source Data List using the Mouse. Please select Reference Numbers 2 and 3 and then press <RETURN>. When the editor appears, type in a definition and description of the item and then press <END>. Now, please follow the same procedure for the following Attribute Class Items:

NOTE: Item and Contributor's Names must be entered with no spaces (such as Your-Name) unless they are enclosed in quotes (such as "Your Name").

7.

Attribute Class Number: 1
Attribute Class Name: **"Item Id"**
Contributor: Your Name
Source Data Reference Numbers: 2 6

Description:

Attribute Class Number: 2

Attribute Class Name: **Amount**

Contributor: Your Name

Source Data Reference Numbers: 4 3

Description:

Attribute Class Number: 3

Attribute Class Name: **"Delivery Date"**

Contributor: Your Name

Source Data Reference Numbers: 13

Description:

Attribute Class Number: 4

Attribute Class Name: **"Department Name"**

Contributor: Your Name

Source Data Reference Numbers: 15

Description:

Attribute Class Number: 5

Attribute Class Name: **"Department Number"**

Contributor: Your Name

Source Data Reference Numbers: 15

Description:

Attribute Class Number: 6

Attribute Class Name: **"Department Size"**

Contributor: Your Name

Source Data Reference Numbers: 14

Description:

Attribute Class Number: 7

Attribute Class Name: **"Employee Name"**

Contributor: Your Name

Source Data Reference Numbers: 12

Description:

Attribute Class Number: 8

Attribute Class Name: **"Employee Number"**

Contributor: Your Name

Source Data Reference Numbers: 0

Description:

Attribute Class Number: 9

Attribute Class Name: **"Item Description"**

Contributor: Your Name

Source Data Reference Numbers: 6

Description:

Attribute Class Number: 10

Attribute Class Name: "Part Number"

Contributor: Your Name

Source Data Reference Numbers: 2

Description:

8. Now, we will go back to the Model Builder to build the diagram. Do so by typing:

- **Exit Data Dictionary** <RETURN>

Next, type: **Create Model** (IDEF0 or IDEF1 [Default IDEF0]) **IDEF1** (Name of Non-existent Model) **Purchasing**

NOTE: This is case sensitive!

Press <RETURN>, and you will see:

Does this model currently exist within a Data Dictionary Project File?
(Y or N)

Type: **Y** You will now see:

Enter the name of the project:

Type: **ABC-Manufacturing**, remembering that this is case sensitive.

9. Now, you can set the name of the view by typing:

- **Rename View** (Parent Model for View to Rename [default Purchasing]) **Purchasing** (A View [default Purchasing (initial-view)] **Purchasing** (initial-view) (New Name for View) **Global** <RETURN>.

10. Now, we will create Entity Classes. First create the entity class, Vendor.

- **Create Entity Class** (Name of Non-existent Entity Class) **Vendor** <RETURN>.

NOTE: The name must be entered either as a string (i.e. typed inside of a pair of double quotes) or with no spaces.

At this point the top left corner of the entity class box will appear on the screen. Place this box as near the upper left corner of the screen as possible. In the model we will place the first four of these entity classes in a line from left to right. Place the next three in a row under these, and the last one on the third row.

Next we will create the entity class, *Purchase Order*.

- Create Entity Class (Name of Non-existent Entity Class)

If, at this time you press Mouse-R, a menu containing all Entity Class Elements not previously used in the model will appear. From this menu, use the mouse to select the entity class *Purchase Order*. Press <RETURN>.

Place the entity class box adjacent to the Vendor box. Using one of the two methods given above create the remaining entity classes in the diagram: *Purchase Order Item*, *Part*, *Department*, *Buyer*, *Shop Order*, and *Employee*. If at this time you return to the Data Dictionary Editor you will notice that the Entity Class Pool has been updated to include all of the new entity classes that you have added.

11. Now, we will create Attribute Classes. Type:

- Create Attribute Class (Name of Non-existent Attribute Class)

If, at this time you press Mouse-R, a menu containing all Attribute Class Items in the Attribute Class Pool appear. From this menu, use the mouse to select the element, "*Delivery Date*." At this time you will be asked for the entity class for which this item is an attribute.

- Move the mouse cursor until the entity class *Purchase Order* has been highlighted. Click the left mouse button to select and <RETURN> to accept the entry.

If you don't wish to use the mouse method, or the Attribute Class hasn't already been created, you can type in the Attribute Class.

- Create Attribute Class (Name of Non-existent Attribute Class) "*Item Quantity*"

The entity class *Purchase Order* will still be selected as described above.

Using one of these methods to create the following Attribute Classes:

12. Attribute Class Item: "Department Name"
Entity Class: Department

Attribute Class Item: "Department Number"
Entity Class: Department

Attribute Class Item: "Department Size"
Entity Class: Department

Attribute Class Item: "Employee Name"

Entity Class: **Employee**

Attribute Class Item: "Social Security Number"

Entity Class: **Employee**

Attribute Class Item: "Item Description"

Entity Class: "Purchase Order Item"

Attribute Class Item: "Item Id"

Entity Class: "Purchase Order Item"

Attribute Class Item: "Order Date"

Entity Class: "Purchase Order"

Attribute Class Item: "Part Number"

Entity Class: **Part**

Attribute Class Item: "Part Type"

Entity Class: **Part**

Attribute Class Item: "Purchase Order Item Number"

Entity Class: "Purchase Order Item"

Attribute Class Item: "Purchase Order Number"

Entity Class: "Purchase Order Item"

Attribute Class Item: **Vendor**

Entity Class: "Address"

Attribute Class Item: "Vendor Number"

Entity Class: **Vendor**

13. Add attributes to the key classes. When the Entity classes were created the system automatically created an empty key class for each of the entity classes. With this command we will add attributes to one of those key classes. To the "Purchase Order.()" key class we will add the attributes "Purchase Order Number" and "Order Date." There are two prompts that must be responded to:

(A Key Class): Select the key class "Purchase Order.()" by Clicking with the right mouse button for a list of completions. Select the key class with a left mouse click. Press SPACE BAR.

(One or more Attributes): Insert the Attribute Classes "Purchase Order Number" and "Order Date". by Clicking with the right mouse but-

ton for a list of completions. Select each of the attribute classes with a left mouse click. Press **SPACE BAR**.

14. [NOTE]: Instead of using the right mouse click to select the above completions you can do the following: Click Mouse-M on *Department* then type: **Add Attributes to Key Class** (A Key Class) From the window on the lower left of the screen select "O". For the prompt One or more Attribute Classes, select with Mouse-L the Attribute Classes "*Purchase Order Number*" and "*Order Date*."

Continue this process until each of the key classes have been populated. You may choose to add additional attribute classes if those that have already been created do not appear to be appropriate for the entity's key class. In particular, add Attribute class "*Vendor Number*" to the Vendor key class.

15. Now we will create Link Classes. You must respond to four or five prompts to complete the insertion of a link class. After responding to a prompt the next prompt is displayed when you press the space bar. The prompts are:

Name of non-existent link class: To this prompt you will type in a string that reflects the business rule described by the link class.

Front Entity Class (An Entity Class): You are being prompted for the independent entity class associated with this link. This entity class may be selected from the list that would be displayed on a right mouse click or moving the mouse arrow over the chosen entity class in the diagram and clicking with the left mouse button.

Back Entity Class (An Entity Class): You are being prompted for the dependent entity class associated with this link. This entity class is selected in the same way that the Front Entity Class is chosen.

Link Cardinality: You will have three choices for this response they are:

- a. [Strong many-to-one]
- b. [Weak many-to-one]
- c. [One-to-one]

Key Class to Inherit: You receive this prompt only if the link cardinality was "One-to-One". The inherited key class must be a key class of the front entity class.

16. The Entity Class Vendor has a strong many-to-one relation to the Entity Class "*Purchase Order*". The relation is "Processes." Type:

- **Create Link Class**

In order, the responses to the prompts will be: **"Processes"** **"Vendor"** **"Purchase Order"** **"strong many-to-one"** Press <RETURN>.

Use a Mouse-M click on the Entity Class **"Purchase Order"** in the model diagram. You will notice that the Entity Class **"Purchase Order"** inherited the attribute **"Vendor Number"** in the **"Vendor"** key class.

17. At this point the model is not complete. You may continue to populate the model with attribute classes, entity classes, link classes, etc as you feel are appropriate. You will use the same procedures that have been described in this section.

4. MDSE Commands

4.1. Command Summary

This section contains an index of all MDSE commands and the page number on which each appears. The commands are grouped according to the overall Model Builder commands, the IDEF0 commands, the Concept Editor commands, the Node Editor commands, the IDEF1 commands, and the Dictionary Editor commands. This provides a handy reference for quickly looking up commands and their use.

Model Builder Commands

47

- | | |
|--------------------------------|----------------------|
| • Create Model | Delete Model |
| • IDEF1 Data Dictionary Editor | Load Model |
| • Refresh | Save Model |
| • Scroll Window | Select Configuration |
| • Select Model | Select Pane |

IDEF0 Commands

50

- | | |
|------------------------|-----------------------|
| • Check Decomposition | Clear Command History |
| • Clear Output History | Concept Editor |
| • Create Activity | Create Concept |
| • Create Relation | Delete Activity |
| • Delete Concept | Delete Relation |
| • Edit Concept | Edit Diagram Text |
| • Edit Glossary Entry | Edit Model Context |
| • Edit Model Purpose | Edit Model Viewpoint |
| • Move Activity | Node Editor |
| • Print Diagram | Print Entire Model |

- Print Node Tree
- Rename Concept
- Show Concept
- Show Decomposition
- Rename Activity
- Reviewers Comment
- Show Context
- Swap Activities

IDEF0 Concept Editor Commands 52

- Create
- Delete
- Exit
- Where Used
- Change Context
- Edit
- Rename

IDEF0 Node Editor Commands 53

- Create Node
- Delete Node
- Edit Comment
- Exit Editor
- Hide Comment
- Hide Node Decomposition
- Move Node
- Show Comment
- Show Node Decomposition
- Undelete Node

IDEF1 Commands

55

- Add Attributes to Key Class
- Check
- Copy Model Portion
- Create
- Create Attribute Class
- Create Entity Class
- Create Key Class
- Create Link Class
- Create View
- Delete
- Delete Attribute Class
- Delete Entity Class
- Delete Key Class
- Delete Link Class
- Describe
- Edit
- Edit Attribute Class Description
- Edit Key Class Description
- Edit Inherited Attribute Class Description
- Edit Key Class Description
- Edit Link Class Description
- Edit Model Description
- Edit View Description

- Hardcopy Display
- Inspect Attribute Class
- Merge Entity Classes
- Middle Click with the Mouse
- Move Entity Class
- Move Link Class
- Print Reviewers Comments
- Rename View
- Refresh
- Relax Net
- Remove Attributes from Key Class
- Reviewers Comments
- Reviewers Comments on Attribute Class
- Reviewers Comments on an Attribute Class in a Key Class
- Reviewers Comments on Entity Class
- Reviewers Comments on Inherited Attribute Class
- Reviewers Comments on Key Class
- Reviewers Comments on Link Class
- Reviewers Comments on Model
- Reviewers Comments on View
- Select
- Select View
- Set Select State
- Show Details

- Show Statistics

IDEF1 Dictionary Editor Commands 61

- Create Source Data Item
- Change Name of Current Project Data Dictionary
- Create Item for Attribute Class Pool
- Create Item for Entity Class Pool
- Create New Model for Current Project Data Dictionary
- Create New Project Data Dictionary
- Create Source Material Log Item
- Edit Dictionary Item
- Exit Data Dictionary
- Delete All Projects In Data Dictionary System
- Load Project Data Dictionary
- Print Current Project and Model
- Save Project Data Dictionary
- Switch Projects
- Switch Models in the Current Project Data Dictionary
- View other Details of Item

4.2. General Model Builder Commands

The initial set of commands that are available to the user after entering the model builder environment are the same for both IDEF0 and IDEF1. Those commands are as follows:

Create Model: This command allows you to create a model in IDEF0 or IDEF1, it will prompt you for the type of model you wish to create and the non-

existant name of that model. If you are creating an IDEF1 model you will be prompted to enter the name of the project that the model is a part of. If that project is not currently loaded in the Data Dictionary system you will be prompted for the name of the file that contains the project. That file will then be loaded into the Dictionary system. It is always best to have the correct Project Data Dictionary in the Data Dictionary system before creating a model.

Delete Model: This command allows you to delete any model that has been previously created. When this command to the system is issued the system will prompt you for the model that is to be deleted. The default will always be the selected model, but you can always choose to delete any model that is currently loaded. A right click with the mouse will give you a list of all of the currently loaded models and any of them may be chosen for deletion. You will not be prompted to save the model; so you should be sure that the model has been saved prior to deletion.

IDEF1 Data Dictionary Editor: This command allows the user to enter the IDEF1 Data Dictionary Editor environment. The system will restore the current environment when the data dictionary session is terminated. (See the corresponding Exit command on page 52 .

Load Model: The command Load Model allows you to load any model that has been previously saved. It is not necessary to specify what type of model it is (IDEF0 or IDEF1) just enter the file where the model is stored. If you are loading an IDEF1 model and the project Data Dictionary is not already in the Data Dictionary system you will be prompted to enter the name of the file that contains the project Data Dictionary. You will not be allowed to load an IDEF1 model if you do not have a project Data Dictionary for it.

Refresh: Refresh clears the model builder window of any garbage. Sometimes when you delete a relation it will leave a hole in a box and refreshing the screen will clean up your display. To execute this command type: Refresh SPACEBAR RETURN. If this does not redisplay the holes in the model, try using the refresh command specifying the option argument ERASE. This will cause everything on the screen to be deleted and redrawn from scratch. **Save Model:** Using this command allows you to save a specified model to a file. It is important that you save any model that you wish to keep because the system will not automatically save your model when you log out. If you are saving an IDEF1 model you will also be prompted to save its project data dictionary. When saving the IDEF1 model and associated project data dictionary it is advisable to use a similar name. ie. [Model Name]-diagram and [Model Name]-dictionary.

Scroll Window: This command will take you to any part of the window that you wish to view. However, it is easier to use the scroll bars along the sides of the windows.

Select Configuration: allows you to choose between 1, 2 and 3 displays. (You can have as many as three display showing at once.) This allows you to view

three models at once if necessary. Refer to Figure 14. You can tell which pane is active at any time by looking at the model builder status line above the pane, it will tell you which model is currently active (selected). A model can be edited only when displayed in a selected pane.

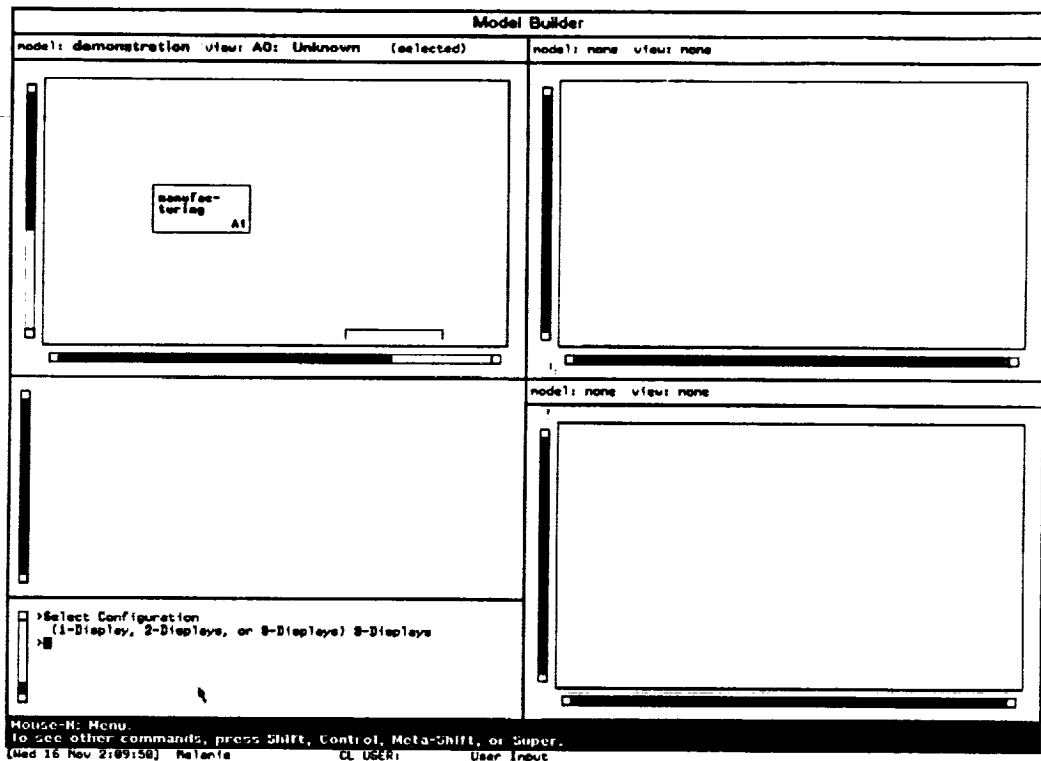


Figure 14. Model Builder with Three Panes Displayed

figure

Select Model: Typing select model allows you to select a model that has been previously created or loaded. If the selected model is an IDEF1 model the Data Dictionary System will have that model's project as the current project. In addition the displayed Entity Class Pool and Activity Class Pool will be the selected models potential Entity Classes and Activity Classes.

Select Pane: This command allows you to switch to another pane by issuing the command and then clicking left on the pane you wish to make active. Only the model in the active pane may be edited.

4.3. IDEF0 Modeler Command Reference

The following is a list of commands for the IDEF0 Model Builder:

Check Decomposition: allows you to check the decomposition of an activity. It will verify that an activity conforms to the IDEF0 methodology. Messages will be displayed which state the number of model elements, and explain the violations. The system pays no attention to these verifications. It is up to you to make the corrections or leave the model as is.

Clear Command History: clears the history of the command window of the model builder.

Clear Output History: clears the history of the text display window of the model builder.

Concept Editor: This command is used to enter the Concept Editor.

Create Activity: This command will prompt for the name of the activity and the activity number that you wish to create. Activity numbers must be either A0 or have the form Adddd..., where d is a number between 1 and 6 inclusive.

Create Concept: will ask you for the name of the concept, or concepts, you wish to create. The concepts will be defined, but no messages will be displayed.

Create Relation: when you create a relation, you are creating a path between one activity and another. You will be asked for the source activity, what concept that activity produces, whether the relationship is an output, input, control, or mechanism, and the destination activity.

- [NOTE]: If CREATE-RELATION throws you into the error handler, (1) press ABORT, (2) type DELETE RELATION RETURN and then (3) type REDO PATHS RETURN. This will verify and correct the model. A full description of the case should be reported to the developers, so that the offending relation may be created.

Delete Activity: deletes any previously created activity that you specify. Be very careful when using this command, since the branch of the tree describing the deleted activity is also deleted.

Delete Concept: deletes any previously created concept, and modifies any relation that is associated with that concept to reflect the fact that the concept has been deleted.

Delete Relation: deletes any previously created relation. The menu notation for DELETE RELATION is: Origin Concept destination reads ORIGIN activity produces CONCEPT as [Input, Control, Mechanism, or Output] of destination activity.

Edit Concept: edits an existing concept. Provides options for changing the name, adding or deleting subtypes or subparts, and adding or deleting concepts which this concept is a subtype or subpart of. Updates all paths to reflect the changes. Be careful when using this command. Due to the inheritance of parts between concepts, "Edit Concept" can be quite destructive.

Edit Diagram Text: edits the text associated with the decomposition of the activity. You will be prompted for the name of an activity.

Edit Glossary Entry: edits the glossary entry for a model element. You will be prompted for the type of model element. The type can be one of the following: ACTIVITY, CONCEPT, or PATH. You will then be prompted for an element of that type to be edited.

Edit Model Context: allows the editing of the text describing the context of the model.

Edit Model Purpose: edits the purpose of the model.

Edit Model Viewpoint: edits the viewpoint of the model.

Move Activity: will allow you to move an activity to another position in the model. For example, say you have activity "Manufacturing Plan" number A32 and you need to move it to A33. Activities may not be moved to a spot already occupied by another activity. A parent cannot be moved to a child diagram.

Node Editor: is a textual representation of the model whereas the standard interface is a graphical representation of the model. You can add activities in the node editor but you cannot see relations or concepts.

Print Diagram: This command will print a hardcopy of the diagram the is currently visible.

Print Entire Model: This command prints the entire model.

Print Node Tree: This command prints an indented list of the activities in the model. It outputs the complete activity number and name to the default print device.

Rename Activity: changes the name of an existing activity. You will be prompted for the name of an activity.

Rename Concept: will allow you to rename any concept that has been previously created.

Reviewers Comment: edits the reviewers comment for a model element. You will be prompted for the type of model element. The type can be one of the following: ACTIVITY, CONCEPT, or PATH. You will then be prompted for an element of that type to be edited.

Show Concept: displays a concept in the text display window. You will be prompted for the name of a concept.

Show Context: allows you to view the context of the current decomposition. This will take you up a level in the heirarchy.

Show Decomposition: allows you to view the decomposition of any activity.

Swap Activities: will allow you to swap one activity with another that has previously been created. For example, if you had manufacturing as A1 and sales as A2, you could switch these activities so that manufacturing would now be A2 and sales would be A1. Relations that you defined previously will remain unchanged, and all effected paths will be redrawn. Again, a child cannot be swapped with its parent.

4.4. IDEF0 Concept Editor Commands

These commands are available within the Concept Editor. Most of these commands are also available from the top level Model Builder. Most top level Model Builder commands are not available from within the Concept Editor. The Concept Editor can only be accessed when the user is editing an IDEF0 model, by executing the command "Concept Editor" on page 50

Change Context: changes the context of the Concept Editor to the selected decomposition. As usual, clicking right will provide a menu of names to choose from. Upon exit from the Concept Editor, the context at the top level will be the same as that of the Concept Editor.

Create Concept: creates one or more top level concepts. "Edit Concept" must be used to add subtypes and subparts. Multiple concepts may be defined by separating the concept names with commas.

Delete Concept: deletes one or more concepts. Use caution! Deleted concepts are not recoverable. Also, remember that inherited parts of the subtypes will be removed when the parent is deleted.

Edit: displays a menu of options for editing the concept definition. Middle mouse click on any concept to bring up this menu. The menu will allow you to change the name, parts, types, and which concepts the concept is a part-of or type-of. Changes to a concept may have drastic effects on the model.

Exit: returns to the Model Builder top level.

Rename: is a quick way to change the name of a concept.

Show Inheritance: lists the inherited parts, and from which parent concepts those parts were inherited.

Where Used: shows the decompositions which use the entered concept. It will not show decompositions where just parts or types of that concept are used.

4.5. IDEF0 Node Tree Editor Commands

These commands are available only from within the Node Tree Editor. The top level Model Builder commands are not available from within the node editor. The Node Editor can only be accessed when the user is editing an IDEF0 model, by executing the command "Node Editor" on page 51

Create Node: This command is used to create an activity from within the Node Tree Editor. The procedure to follow when using this command and the prompts to respond to are as follows:

- **Create Node** <SPACE BAR>
- The prompt is **Create Node** (an Activity Name). The user will type in the name for a new activity. The requirements and restrictions for this name are the same as those for creating an activity from within the model builder. See **Create Activity** on page 50
- Type a space and the next prompt is (an activity number). This number should be a unique activity number.
- After typing another space the screen will display the following: **Create Node** (an Activity Name) **name** (an Activity Number) **number** (Under What Heading).
- The user is expected to select the Context for the activity. The current system will allow the user to select any heading for the new activity. For instance you would be allowed to place the activity "A23" under the heading "A33". This will be corrected in later versions, but for now if this mistake is made the best solution is to move the node (See "Move Node" on page 54) to the correct heading.

Delete Node: This command allows the deletion of only those nodes created in the current node editor session. Once an activity has been placed within the model it can not be deleted from within the node editor.

Edit Comment: This command allows the person who is reviewing the activities in the node tree editor to make comments. The commenting may be done either from within the model builder or from the node tree editor. After typing in the command "Edit Comment" the user is prompted for the name of an activity. At this point any activity may be selected with a left mouse click. A ZMACS editor will pop up in the left hand corner of the screen. After typing in a comment, adding to a comment, or just reading the comment the window may be removed by pressing the <END> key. If the comment is a new comment it will remain displayed until it is specifically hidden. See the command **Hide Comment**.

Exit Editor: Once a session with the node tree editor is over this command allows the user to return to the model builder. All nodes that have been created in the editor will be drawn in the model in the appropriate positions.

Hide Comment: When the Node Tree Editor is first entered all reviewers comments will be displayed. This command will allow the user to hide the comments for a specific activity. The comment may still be edited (See the command "Edit Comment." Or by using the command "Show Comment" on page 54 the comment may be redisplayed.

Hide Node Decomposition: Another form for this command is "Hide Activity Decomposition." This command can be activated in two ways. The best way to do this is to move the mouse cursor (arrow) over an activity (node). If the node has a decomposition, a box will appear around the node and the mouse line at the bottom of the screen will display the command "Mouse-L Hide Activity Decomposition". One left click with the mouse will hide the decomposition of the node. The command may also be executed by typing it in followed by the activity number. This method is not as acceptable as using the mouse because it will accept any node even if the decomposition has already been hidden or if it does not have a decomposition. This command is useful in reducing the size of the display. See the companion command "Show Node Decomposition."

Move Node: This command will allow the user to move any node that was created within the current Node Editor session to another heading.

- The first prompt is for the Activity to move. This activity can be selected with Mouse-L.
- The next prompt is for the heading under which the activity is to be placed.
- The final prompt is for the new activity number.

Show Comment: Any reviewer's comment that was previously hidden may be redisplayed on the screen by using this command. The user is prompted to enter the name of the activity.

Show Node Decomposition and Show Activity Decomposition: At the time that the Node Tree Editor is entered all activities and their decompositions are visible. Using the command "Hide Node Decomposition," various activities may have had their decompositions hidden. The command Show Node Decomposition will cause the decompositions to be redisplayed. There are two ways to activate this command. The best way to do this is to move the mouse cursor (arrow) over an activity (node). If the node has a hidden decomposition, a box will appear around the node and the mouse line at the bottom of the screen will display the command "Mouse-L Show Activity Decomposition". One left click with the mouse will redisplay the decomposition of the node. The command may also be executed by typing it in followed by the activity number. This method is not as acceptable as using the mouse because it will accept any node even if the decomposition is already visible or if there is not decomposition to show.

Undelete Node: A node may only be deleted if it has been created within the current Node Editor session. However, the node editor will remember all nodes deleted in the session and any of them may be returned to the model by executing this command. The user will be prompted for the name of the node to restore. A right mouse click will provide a list of all deleted nodes and a left mouse click on any nodes in the list will allow the node to be returned to the node tree in its previous position.

4.6. IDEF1 Modeler Command Reference

Add Attributes To Key Class: Any attribute class in an entity class either owned or inherited may be added to the entity class's key class in response to this command. Pressing the help key in response to the "attribute to add" request will provide a mouse selectable menu of available attribute classes. The user will not be allowed to add attribute classes to the key class that would result in a non-unique key class.

Check: There are two completions to the command that begins with "check". They are **Check it out** and **Check for increasing cycles**. "Check it out" generates English like statements of all possible implications in the model. These are placed in a menu in which the reviewer can select a choice of "agree," "disagree," or "don't know."

Copy Model Portion: This command is used when creating a model by merging two existing models that are known to be correct. The modeler is allowed to copy all or part of one of the two models into a new model. When copying a partial model inherited attributes may extend the partial model copied or become owned.

Create: The six completions to this command are used to create new models and model elements. This command enforces the IDEF1 language rules, i.e. the modeler will not be allowed to create an entity class if another entity class already exists within the model by that name.

1. **Create Attribute Class:** When creating an attribute class, the first thing that the system will prompt you for is for a name of a non-existing attribute class.
 - Mouse-M will display a menu of the potential attribute classes that are stored in the project data dictionary. One of these may be chosen by placing the x arrow over the chosen attribute class name, and clicking Mouse-L, or
 - You can type in the attribute class name.
 - You will then be prompted for the entity class that owns that particular attribute. The entity class may be selected with Mouse-L on an entity class in the selected model or, with Mouse-

R, the system will give you a menu of all available entity classes in the currently selected model. Using command completion, you may partially type in the name of the entity class and the system will complete the selection.

2. **Create Entity Class:** First you are prompted for the name of an entity class. There are two ways to get the name of the entity class:

- Mouse-M will display a menu of the potential entity classes that are stored in the project data dictionary. One may be chosen by placing the x arrow over the chosen entity class name, and clicking Mouse-L, or
- You may type in the entity class name.

The name of the entity class must be unique. If it is not, it will not be added.

3. **Create Key Class:** The system will prompt you to provide the name of the entity class that will be required for this key class.
4. **Create Link Class:** The system will prompt you to provide the name of the link class that should be in the form of a string. For example, 'has.' Then you will be prompted for the front entity class, which is the independent entity class. This entity class may be selected by moving the mouse cursor over entity class on the screen and clicking Mouse-L or by typing in the name. You will be then prompted for the back entity class, or the dependent entity class. This may be selected in the same manner as the front entity class described above. Next the prompt is for the link cardinality. The link cardinality may be one-to-one, strong-many-to-one, or weak-many-to-one. This may be typed in or a Mouse-R click will give you a menu which will list all link cardinality choices available and you may choose from this menu by placing the 'x' over the selection and clicking Mouse-L. If the choice was one-to-one, you will receive an additional prompt for the key class to inherit. Mouse-R will give you a list of key classes that will be inherited and they may be selected in the same way as the selection of the link cardinality or by typing it in. Mouse selection is preferred.
5. **Create Model:** Inherited command from the General Model Builder Commands. See page 47
6. **Create View:** Multiple views may be created in a selected model with this command. The user is first prompted for the name of the model in which the view is to be created. A right click with the mouse will result in the display of those models that are available.

A model may be selected from this list by moving the mouse arrow over the name of the chosen model and clicking left with the mouse. Next the user is prompted for the name of the new view to create.

Delete: Model elements may be deleted with the proper completions to this command. All dependencies are automatically updated once this command is executed.

1. **Delete Attribute Class:** The system will prompt for an owned attribute class. There are three ways to get this:
 - You may type the attribute class name in,
 - Mouse-R will bring up a menu of all owned attribute classes. One may be selected by clicking Mouse-L, or
 - If all you know is the entity class name, you can click Mouse-L on the entity class on the screen and, in the text display window, you will see the entity class description. From this entity class description, you can select the attribute class for that particular entity class.

Only owned attribute classes can be deleted.
2. **Delete Entity Class:** The system will prompt for the name of an entity class. At this point, the best method for selecting the entity class to delete is to select it with the mouse. Move the mouse cursor over the entity class to be removed, click Mouse-L, and then press <RETURN>. There are also other ways to select this entity class, but this appears to be the most secure.
3. **Delete Key Class:** The system will prompt for a key class name. This name should be unique and may be selected from the menu that pops up from a Mouse-R click. The best method is to use the Show Details command (see page 60) and select the appropriate key class from its entity class description.
4. **Delete Link Class:** The system will prompt for the name of the link class. The best method is to use a right mouse click that will display a menu of possible completions. Another possible method is to highlight the link cardinality indicator on the link class and click Mouse-L. This will give you the completion and allow the deletion of that link class. When the link is deleted, all attributes and key classes that are inherited through that link will be automatically removed from the dependent entity classes.

5. **Delete Model:** Inherited command from the General Model Builder Commands. See page 48

Describe: This system command will provide the system description of any model element.

Edit: Associated with each model element, including the model and model view is a description. The system has only one editor window that is reused for all model elements. The execution of the "Edit-[model element type]-Description" command will result in the display of the editor window and any previous description associated with the named model element. Each of the eight (8) completions for this command will provide a small editor window in which the model element description can be inspected, edited or inserted. In each command the user may click the right mouse button to get a menu containing a list of the elements whose descriptions may be edited. Listed below are the IDEF1 model elements whose descriptions may be entered by the modeler.

1. **Edit Attribute Class Description:** After typing the command the user will be prompted to enter an attribute class.
2. **Edit Attribute Class in Key Class Description:** After typing the command the user will be prompted to enter an Attribute class in a key class.
3. **Edit Entity Class Description:** After typing in the command the user is prompted to enter an entity class to edit.
4. **Edit Inherited Attribute Class Description:** After typing in the command the user is prompted to enter an inherited attribute Class to edit.
5. **Edit Key Class Description:** After typing the command the user is prompted to select a key class to edit.
6. **Edit Link Class Description:** After typing the command the user is prompted to enter a link class to be edited.
7. **Edit Model Description:** After typing the command the user is prompted to enter the name of a model description to edit.
8. **Edit View Description:** After typing the command the user is prompted to enter the view to edit.

Hardcopy Display: The graph of the model will be printed by this command

Inspect Attribute Class: This command will print the system description of an attribute class. This description includes all that the system stores concerning the requested attribute class.

Merge Entity Classes: Two entity classes that exist in the same pane become one when this command is executed. The modeler determines which entity class is to remain. All attribute classes and key classes of both entity classes become part to the merged entity class.

Mouse-M: If the middle mouse button is pressed once on a highlighted entity class, that entity class's dictionary items. These items include:

- The Entity Class's Unique Name,
- The key class for the entity class and the attributes in the key class, and
- The unique name for all of the attribute classes both owned and inherited.

See the command Show Details on page 60

Move Entity Class: This command allows to modeler to move the graphical display of an entity class to another position in the model. To complete this command the user selects an entity class with Mouse-L and presses <RETURN>. This selects the entity class that is to be moved. The mouse cursor becomes the left corner of the entity class box. Move this cursor to the new location that the entity class is to be positioned and click Mouse-L a second time to place the entity class. The system will adjust the positioning of all links. Should some line remain on the screen this can be corrected by using the command Refresh. (See Refresh on page 48)

Move Link Class: Using this command the user can manually reposition the links between two entity classes. The link to be moved is selected with Mouse-L. After pressing the return key a dot will appear on the independent entity class. This dot will follow the mouse cursor around the entity class box. Select the position on the entity class where the link is to start and click once with the Mouse-L button. Next a dot will appear on the dependent entity class. Move the mouse until the dot is in the position that you wish to have the link join the entity class and click Mouse-L again.

Print Reviewers Comments: Using this command the user can produce a hard-copy of the comments that the reviewers of the model have made. The comments are entered by executing the command described on page 60 .

Relax Net: This command causes the current model to be restructured to optimize the layout for minimal line crossings.

Remove Attributes From Key Class: Execution of this command is allowed on an entity class's owned attributes and the inherited attributes that have been added to the key class. The system will not allow the modeler to remove those attributes that would create a key class that was not unique.

Rename View: The execution of this command allows the user to change the name of any view in a model. The user is first prompted for the name of the model. Mouse-M will result in the display of those models that are available. A model may be selected from this list by moving the mouse arrow over the name of the chosen model and clicking Mouse-L. Next the user is prompted to enter the view to be renamed. Mouse-R will provide a list of the views in the selected model. A view may be selected from this list by moving the mouse arrow over the name of the chosen view and clicking Mouse-L.

Reviewers: There are eight(8) completions for this command. This command would be used by someone who was reviewing an already completed model. It uses the same editor window that is used by the edit command, but in this case it contains the reviewer comments on a particular model element. After typing in one of the following commands a right mouse click will provide a menu of acceptable completions for the command.

1. Reviewers Comments on Attribute Class
2. Reviewers Comments on an Attribute Class in a Key Class
3. Reviewers Comments on Entity Class
4. Reviewers Comments on Inherited Attribute Class
5. Reviewers Comments on Key
6. Reviewers Comments on Link Class
7. Reviewers Comments on Model
8. Reviewers Comments on View

Select: There are three(3) completions for this command. The modeler may select a configuration, pane, or view. Selection of a configuration means that the user chooses to have one window, two windows or three model windows. A different model may be displayed in each pane (window). Pane selection allows the user to choose the pane in which to perform operations.

Select View: A particular view of a model can be selected to be displayed in a pane with the "Select View" command.

Set Select State:

Show Details: A command that is available with a middle mouse click. Allows you to show details of any entity class in any model that is visible on the screen. This is available with any command but it is not a command completion. For example, if you are deleting an attribute class, before the attribute class is input, you can click middle mouse button on any

entity class and information on the entity class will be displayed in the text display window. The details of the descriptive information is: the entity class name, the key classes, all attributes in key classes, and all attribute classes that are either owned or inherited. A command requiring any of these elements for a completion, can be completed with a left mouse click on the appropriate element from this description.

Show Statistics: This function implements the semantic matching support for model integration. Entity, key, attribute, and link classes are checked for possible merger, or subsumption after a copy model portion command is executed. Suggestions are made based upon a statistical threshold which may be chosen by the user.

4.7. IDEF1 Data Dictionary Editor Commands

The IDEF1 Data Dictionary Editor is entered by executing the command IDEF1 Data Dictionary Editor (see page 48). The commands that are available to you for editing the Data Dictionary are as follows:

Change Name of Current Project Data Dictionary: This command allows you to change the name of a Project Data Dictionary. The use of this command can cause problems if the project is already associated with a model diagram. The reason for this is that each model diagram knows the name of the project that it is a part of. The primary purpose of the command is to correct typing and spelling errors when the project data dictionary is originally created.

Create Item for Attribute Class Pool: This command is used to create a potential attribute class. You will be prompted for a unique attribute class number, the name of the person creating the attribute class, and a list of one or more number of source data items in which this attribute class is referenced. These source data numbers must be selected from the Source Data List and are selectable only with the mouse. Pressing the space bar indicates to the system that the response to a prompt is complete; so the names should be entered with dashes instead of spaces or typed inside double quotes. The completion of the information is indicated by pressing return. After completion of this information a editor window pops up in which the attribute class can be described. This editor is a full ZMACS editor. You indicate that the editing in the window is complete by pressing the END key on the keyboard. If this attribute class eventually becomes part of a model diagram this description will be available in the diagrammer. An attribute class has an owner (an entity class) and if this item becomes an attribute of an entity class in a model diagram the owner will be added at that time.

Create Item for Entity Class Pool: This command is used to create a potential entity class. You will be prompted for a unique entity class number, the name of the person creating the entity class, and a list of one or more

number of source data items in which this entity class is referenced. Completion of each of these prompts should be followed by pressing the space bar to cause the next prompt to appear. The source data numbers must be selected from the Source Data List and are selectable only with the mouse. After completion of this information press return and the editor window pops up in which the entity class can be described. This editor is a full ZMACS editor. You indicate that the editing in the window is complete by pressing the <END> key on the key board. If this entity class eventually becomes part of a model diagram this description will be available in the diagrammer. The command should not be used unless the Project Data Dictionary has an Entity Class Pool, Attribute Class Pool pair.

Create New Model for Current Project Data Dictionary: This command is used to create an empty Entity Class Pool and Attribute Class Pool. To avoid errors this command should be used as soon as a new project is created. The term Model used here is not a model diagram. It refers to the set of potential entity classes and potential attribute classes for the diagram. No commands should be executed by the user until the Data Dictionary System has at least one Project Data Dictionary and one Model.

Create New Project Data Dictionary: This command is used to create an empty Project Data Dictionary. As soon as the new project has been created one or more models should be created in the Project Data Dictionary. No commands should be executed by the user until the Data Dictionary System has at least one Project Data Dictionary and one Model.

Create Source Material Log Item: Each source material log item will be added to the system by using this command. The user is prompted to enter the name of the item. This name should be typed without spaces or typed inside double quotes. Press the space bar and you will be prompted for the source number, press the <SPACE BAR> and type the name of the person who collected this source material item. Press Return. After completing this entry an editor window pops up to allow the entry of the item description. This editor is a full ZMACS editor. You indicate that the editing in the window is complete by pressing the <END> key on the key board.

Create Source Data Item: A source data item is added to the list using this command. You will first be prompted to enter the name of the data item. This name could be typed without spaces or typed inside double quotes. Press the space bar and you will be prompted to enter a unique number to associate with the item, press space after this entry to enter the name of the person who selected this data item. After typing in the name press the <SPACE BAR> again. Now using the mouse you will select from the Source Material Log those items which refer to this data item. These numbers will be automatically be placed in a list. Press return to complete the addition of the data item to the list and an editor window will appear in which the description can be included. This editor is a full ZMACS editor. You indicate that the editing in the window is complete by pressing the <END> key on the key board.

Delete All Projects In Data Dictionary System: This command should be used with extreme care. **DO NOT USE THIS COMMAND UNLESS YOU WISH TO REMOVE ALL OF THE PROJECT DATA DICTIONARIES IN THE DATA DICTIONARY SYSTEM.** The models in the model diagrammer that are associated with the projects that you are deleting should be **SAVED** and **DELETED** before you execute this command in the Data Dictionary System!

Edit Dictionary Item: This command may be executed by holding down the <HYPER> key and pressing Mouse-M on the dictionary item that you wish to edit. Another way to execute the command is to type the command on the keyboard then select the item to edit with Mouse-L. Use this command if you wish to update or change some of the information that you are keeping about any item in the Project and Model that are visible on the screen. The values associated with the selected item appear at the bottom of the screen. These values may be changed by selecting a value with Mouse-L. This will cause the current value to disappear and you may type in the new value. Currently you are not allowed to change the owner of an attribute class. The owner of an attribute class is set by the model diagram builder when the attribute is placed in an entity class. You should not change the item's number since the system has already forced it to be unique. If you wish to change the reference list associated with the item, the selection of sources must be done with the mouse.

Exit Data Dictionary: This command is used to exit a project data dictionary editing session and returns the user to the model builder environment. The dictionary system remembers the configuration of the model building environment and the execution of this command restores that configuration.

Load Project Data Dictionary: This command is used to load a project data dictionary that has been saved in a previous session. The full pathname should be typed in.

Print Current Project and Model: This command will ask for confirmation (Yes or No). If the response is "Y" <RETURN> the system will output to the default print device a copy of the Source Material Log, Source Data List, Entity Class Pool, and Attribute Class Pool of the current project and model.

Save Project Data Dictionary: The system will not prompt you to save your files; therefore you should remember to save all project data dictionaries that you are working on before you quit a session. You should type the full pathname for all files that you save. It would be a good practice to establish a set naming convention for all data dictionary files that would allow you to remember them and that would allow you to associate them with the proper model diagram files. ie. To all of my IDEF1 diagram files I append "-diagram" to all of my dictionary files I append "-dictionary". Usually the first part of the name of the files will be identical.

Switch Models in the Current Project Data Dictionary: If the currently visible project has multiple models associated with it this command will allow you to display and work on another model. The command causes a menu of names of selectable models to appear on the screen. Left click with the mouse to choose one of the models.

Switch Projects: If you have more than one project data dictionary loaded in any session this command will allow you to select any of the project data dictionaries to display and edit. The command causes a menu of names of selectable projects to appear on the screen. Click Mouse-L to choose one of the models.

View Other Details of Item: This command may be executed by holding down the <META> key and pressing the Mouse-M on the dictionary item that you wish to edit. Another way to execute the command is to type the command on the keyboard and select the item with Mouse-L. Executing this command allows you to look at all of the information about any item currently visible on the screen. The screen may be removed by moving the mouse (it appears as a circle inside the temporary window) outside the window and clicking Mouse-L.

5. Model Element Dictionary Development User's Manual

Introduction

The Model Element Dictionary Developer (MEDD) is a prototype software application that was developed by Knowledge Base Engineering Enterprises (KBEE) under a contract to Texas A&M University, Texas Engineering Experiment Station, for the "Knowledge Based Integrated Information Systems Development Methodologies" project. The MEDD provides an interactive platform from which to specify and maintain function, information and enterprise model element data. The MEDD is a tool that is to be used during information systems engineering activities. It enables the user to capture, categorize and output model element data that is obtained during interviews with engineering and manufacturing operations personnel. The MEDD is designed to operate on 100% IBM PC compatible computers including laptops, portables and personal workstations. This means that the MEDD can be used on the shop floor as well as in the office or even on an airplane if desired.

5.1. MEDD Development Platform

The MEDD is written in the Q-PRO fourth generation application development language. The underlying database utilizes a B+ Tree File System. The application driver is a Microsoft Mouse. The mouse is used to make menu selections for chaining to forms and invoking form/database functions, both of which are accessed via pull-down menus. Character strings (two alphabetic characters) or function keys (F1-F19) are identified in the pull-down menus for each MEDD menu selection so that the user can choose either the mouse or the keyboard for menu selection/function invocation. The menu bars, pull-down menus and pop-up help windows were all developed using the Microsoft Mouse Menu Programming Language.

5.2. MEDD Architecture

The MEDD system architecture consists of three subsystem modules: an application tutorial, a dictionary developer, and an output utility. The tutor enables the user to step through a hierarchy tree that graphically illustrates the MEDD menu structure. At each node in the tree, the user can obtain a definition via a pop-up help window that explains the purpose of the menu selection. The dictionary developer provides the user with interrelated data entry/retrieval forms for specifying and maintaining model element data in a database. Form data is retrieved from, manipulated and entered into the database through the use of pull-down menus.

The output utility enables the user to queue up to 16 report files, database files and/or sequential (export) files in a print spooler for output to the display screen or a print device. The sequential files can be used as input to other applications such as the automatic model diagram layout tool being developed by Texas A&M as part of this project.

5.3. Getting Started with the MEDD

The MEDD requires little or no formal documentation or training beyond the standard IDEF method training. It provides pull-down menus, pop-up help windows and data entry/retrieval forms for specifying and maintaining the dictionary information for following model elements:

- **Function Model Elements**

- Function Definition form enables the user to specify and maintain the function name (a function is anything that can be named with an active verb phrase, for example: assemble part, classify test results, resolve discrepancy, etc.) and the parent function name (next higher level function related to the named function), to define the purpose of the function, and add any comments that may be necessary.
- Function Inputs
- Function Controls
- Function Outputs
- Function Mechanisms
- Interface Definition
- Interface Attributes
- Function Sequence
- Function Decomposition
- Function Locations

- **Information Model Entity Class Elements**

- Source Material Log
- Source Data List

- Entity Class Pool
- Entity Class Definition
- Entity Class Synonyms
- Relation Class Definition
- Node Cross Reference
- Attribute Class Pool
- Attribute Class Definition
- Attribute Class Migration Index
- Attribute Class Cross Reference
- **Enterprise Model Elements**
 - Project Pool
 - Process Pool
 - Life Cycle Pool
 - State Pool
 - Life Cycle States
 - Group Pool
 - Location Pool
 - Group Locations
 - Domains
 - Domain Values
 - Access Reasons
 - Revision Reasons
 - Notifications
 - Approvers

The system is designed to essentially replace the manual forms used in the standard IDEF0 and IDEF1 modeling disciplines. All of the standard dictionary information forms are provided. Based on the experience of IDEF modelers and systems analysts participating in this effort, several additional forms are provided. These optional forms should be viewed as "authors" conventions and not as approved extensions to the IDEF methods. The system has a built in tutorial support to provide method information in addition to the help information that supports the use of the tool.

5.4. Summary of Commands for the MEDD

This section provides a description of the commands required to navigate through the system and enter and retrieve data from the MEDD.

MEDD MENU Commands

Main Menu Bar: The following menu items will appear along the top of the terminal screen upon successful log-in to the MEDD application:

- METHODS
- TUTOR
- DIRECTORY
- EXIT
- MEDD

In order for the user to make selections from this main menu bar, press Mouse-L once. The cursor will move up and by moving the mouse horizontally, the user can position the cursor over the desired menu selection. When the cursor is positioned over a menu selection, the character block on the screen is changed to inverse video. To make a menu selection, the user must press Mouse-L.

Main Menu Pick Definitions

- **METHODS** - Enables the user to access the METHOD drop-down menu which consists of the following menu picks:
 - **Function Model**- Enables the user to access to the FUNCTION MODEL drop-down menu.
 - **Information Model** - Enables the user to access to the INFORMATION MODEL drop-down menu.

- **Enterprise Model** - Enables the user to access to the ENTERPRISE MODEL drop-down menu.

Selection of any of the above menu items will display a menu that contains a list of related forms that can be accessed and used for data entry/retrieval.

- **TUTOR** - Enables the user to leave the dictionary developer and enter the application tutorial.
- **DIRECTORY**
- **EXIT** - Enables the user to access the EXIT drop-down menu which consists of the following menu picks:
 - **To DOS** - Exit to Disk Operation System.
 - **To OUTPUT** - Exit to Output Utility.
- **MEDD** - Enables the user to access to the MEDD pull-down menu which displays the following menu items:
 - **MEDD Info ...** - Application name, version and authors.
 - **Date** - Current date.
 - **Time** - Current time.

Form Menu Bar: The following menu will be encountered upon successful entry into a form:

- **EDIT**
- **FORM**
- **HELP**
- **EXIT**

In order for the user to make selections from this form menu bar, press Mouse-L once. The cursor will move up and by moving the mouse horizontally, the user can position the cursor over the desired menu selection. When the cursor is positioned over a menu selection, the character block on the screen is changed to inverse video. To make a menu selection, the user must press the left button on the mouse.

Form Menu Item Definitions

- **EDIT:** - Enables the user to access to the EDIT drop-down menu which consists of the following menu picks:
 - **List Oriented Forms**
 - **Add Record(s)** - Add record(s) to the database. If the user is in the Function Sequence form the following menu picks will be displayed when the Add Record(s) menu pick is selected:
 - **First Time Entry** - Add record(s) in a specific sequence for the first time.
 - **Insert Before** - Inserting a function before another function in the sequence.
 - **Insert After** - Inserting a function after another function in the sequence.
 - **Delete Record(s)** - Delete record(s) from database. If the user is in a list oriented form the following menu picks will be displayed when the Delete Record(s) menu pick is selected:
 - **Instance at Current Cursor Position** - Delete a record instance based upon the current cursor position in the list of record instances.
 - **All Displayed Instances** - Delete all record instances that are currently displayed in the list of record instances in the form.
 - **Definition Oriented Forms**
 - **Add Record** - Add one record to the database.
 - **Modify Record** - Modify one record in the database.
 - **Delete Record** - Delete one record to the database.
- **FORM** - Enables the user to access to the FORM drop-down menu which consists of the following menu picks:
 - **Fetch Data** - Retrieve data from the database and display it on the form.

If the user is in the Function Sequence form the following menu picks will be displayed when the Fetch Data menu pick is selected:

- Alphabetically - Retrieve data from the database alphabetically.
- Sequentially - Retrieve data from the database in sequential order.

If the user is in a definition oriented form the following menu items will be displayed when the Fetch Data menu item is selected:

- Previous Record - Retrieve previous record in the database based upon record that is currently displayed in the form.
- Next Record - Retrieve next record from the database based upon record that is currently displayed in the form.
- Specified Record - Retrieve record from database based upon the record key value that is displayed in the form.
- Clear Data - Remove data from the form.
- Print Data - Send form and data that is currently displayed in the form to the print device.
- HELP - Enables the user to access to the HELP pull-down menu which consists of the following menu items:
 - Form Field Definitions- Definitions of the field that are on the form that the user is currently working in.
 - Field Format Definitions - Definitions of the field formats for the form the user is currently working in.
- EXIT - Enables the user to select the EXIT pull-down menu which consists of the following menu items:
 - To Related Form - Chain to form that is related to the form that user is currently working in. If the user selects this menu item, the EXIT TO RELATED FORM menu will be displayed. This menu enables the user to pick which form to chain to (based upon currently displayed form).
 - To Generate ... - Generate report and/or sequential file. If the user selects this menu pick the GENERATE ... menu will be displayed. This menu enables the user to pick between report generation and sequential file generation (depending upon currently displayed form).

- To Main Menu - Leave form and return to the Main Menu Bar.
- To DOS - Leave form and return to Disk Operation System.

6. Micro-Based Fact Collection Tool User's Manual

Introduction

The Fact Collection Tool is designed for use by an analyst to assist in the collection and organization of data gathered during the interview processes associated with a system analysis and modeling task. The basic concept behind the development of this tool is that during the interview process an analyst is presented with assertions of facts by the domain expert. The analyst also makes observations of the domain (which again take the form of facts observed). These facts need to be collected and preserved in such a way as to allow them to serve as the basis for a number of decision making processes throughout the system development process. Thus, this tool can be thought of as a computerization of the analyst's notebook. As such, the primary mode of entry of information into the FCT is through the entry of data in the form of "sentences" (capture of "Dialog"). The analyst user of the FCT can enter these assertions and then classify portions of the text as references to instances of ten general concept types. This allows the user to capture both named and descriptive references to the concepts in the domain. Once a reference has been captured the tool supports the acquisition of additional information about that concept. This support is provided through a set of standard data collection forms associated with each concept type. For example if a named or descriptive reference is categorized as a reference to a "physical object" the standard form will prompt for such information as the "has parts" and "part of" relations to other physical objects. These back-up data collection forms were composed after a detailed analysis of the practice of expert systems analysts and the interview procedures of a number of system analysis methods. However these collection forms are known to be incomplete. Therefore the FCT allows the capture of additional dialogue (via the dialogue capture mode) during the filling out of a standard data collection form. This support eliminates the need for a general "comments" section on the data collection form, allowing all input to be available for subsequent processing and analysis.

The following user's manual describes the general usage procedures and commands for the prototype FCT developed under this effort. This tool was constructed primarily to investigate the feasibility of providing such collection support on a micro-processor based delivery platform. Therefore, it should not be considered as a production utility. No warranties or guarantees are claimed for its performance or robustness.

6.1. Getting Started

After the FCT has been installed and booted the user will be presented with a top level screen. The system should be run from the computer's hard drive, if one is available. The following sections describe in general the allowable operations at this screen and all subsequent screens.

6.1.1. Main Screen

The main screen is the opening menu. At this time there are only two keys which the DCT will respond to at this screen. They are Ctrl-C (pressing the control key and "C" at the same time) and the escape key, <ESCAPE>. The Ctrl-C sequence will place you in the collection mode. <ESCAPE> will exit the program. In the next version of the IDSE micro-tools Ctrl-A will place you in the fact analysis mode, and Ctrl-M will place you in the MEDD.

6.1.2. Data Collection

The collection section of this tool is designed to assemble data during an interview. It uses key sequences to move from screen to screen instead of menus in order to increase the speed of the tool. The data can be broken down into three categories; the stage of the interview, statements, and classified items.

6.1.3. Setting the Stage

The stage form is displayed by pressing Ctrl-S at the Collection Screen. Headings followed by spaces for data entry will appear to gather information concerning the interview setting. This information includes who the interview is with, where and when it will take place, etc. A segment of one line will be in reversed video (it will have a light background with dark letters). This is the entry prompt and shows you how much space has been allocated to hold the information for the label next to that space. At this time you may move the entry prompt to enter information elsewhere in the form or type in a response for that label.

To move the entry prompt, you must use the cursor keys. These are the group of keys with the arrows on them or the words "Home," "PgUp," "End," and "PgDn." The arrows will move the prompt in the direction specified, scrolling the form up and down if needed. The "Home" and "End" keys will move the prompt to the first and last labels on the form. The "PgUp" (pronounced page up) key will position the form so that the screen displays the labels and entries that were above those currently on the screen. The "PgDn" key displays the screen of labels below the current screen.

To enter information, type it in from the keyboard. Originally, all of the information will be blank, and the first time you type in a piece of information, the blanks will be replaced by your entry. All of the cursor keys perform various functions. The "Home" key will position the cursor at the beginning of the entry space. The "End" key will place the cursor at the end of that space. The "Ins" key will toggle the entry mode between inserting text and overwriting it. And the left and

right arrow keys will move the cursor in the appropriate direction. To have the system accept what you have typed, press the "Enter" key. Or, to disregard what has been typed, press the "Esc" key. This will return that entry to the state that it was in before you started modifying it.

If you find that you have made a mistake, and wish to correct an incorrect entry, move the reverse video prompt to that entry and press "Ctrl-E" to enter the edit mode. This is exactly the same as if you are entering the information for the first time, except that the current information is kept and you are allowed to change it. All of the cursor movement keys will work the same way.

6.1.4. Entering Dialogue Statements

Statements can be entered from the collection screen or after pressing "Ctrl" and a function key to enter the form. To enter a statement you will need to press one of four key sequences. These are "Ctrl-G," "Ctrl-B," "Ctrl-R," or "Ctrl-P." The collection screen has these key sequences listed to remind the user. This will open up a window with the specified category for the sentence. "Ctrl-G" will classify the statement as a general statement. "Ctrl-B" will classify it as a business rule. "Ctrl-R" will signify a relation. "Ctrl-P" will classify the statement as a problem statement. Only one sentence or idea should be entered per window. The maximum allowable length is 200 characters.

While you are typing the statement, you may classify various parts of the statement into the previously listed classifications. Each part that is classified must be less than thirty-six characters long. To classify a string of text, merely press the function key associated with that classification. The text that will be classified will be shown in reverse video. Press the function key again to complete the classification. This text will be stored and will appear on the list of those items as described previously. If you press the control key with the function key then a blank form associated with that key will appear, ready for more information to be entered about that item. When you are through with that form, the statement window will return with the statement just as you were typing it.

6.1.5. Entering Detailed Descriptions

The Collection screen shows a list of classifications in a column. These serve to categorize certain phrases or strings of text. A function key has been assigned to each of the ten different classifications. The classifications are:

- F1 Activity
- F2 Physical object
- F3 Artifact
- F4 Organization
- F5 System

- F6 Process/scenario
- F7 Event
- F8 Link
- F9 Descriptor
- F10 Statement

Note that F10 "statements" only show thirty-five of the 200 characters. Three dots denote truncation if needed. If there are not any characters filling the space to the dots then there is no statement or it did not fill the 35 character space. To obtain a form for entering information about an item, press the control key with the associated function key. Each form is different and has labels similar in structure to the stage form. These are provided to help you obtain all of the needed information. The operation of these forms is just like that of the stage form.

At the collection screen, if you press the function key by itself, a list of items with that classification will appear. You may then select one of those items to view its associated form or enter in more data.

6.1.6. Analysis Mode Overview

(Note: The analysis and modeling sections have not been completely implemented yet.)

From the main screen, pressing "Ctrl-A" will place you in the analysis mode. In this mode you may browse and modify items and statements in several ways. It is similar to the collection mode, where you request a list of items or statements and can select one item from the list and browse or modify it. The second is by stepping through the interview in the order that items were entered. After starting this type of browsing, you may use the cursor keys to display the next or previous item. A third way to browse is to request the display of the various links between the items or statements and other items or statements

Since the collection mode allowed multiple entries of the same item, the analysis mode allows you to combine entries that are actually the same item. This combination will also maintain all of the links from both of the previously separate items. You may also reclassify a misclassified item. For example, if an item was classified as an event that is actually an activity, it may be reclassified to its correct type.

6.1.7. Model Element Dictionary Development Mode

Pressing "Ctrl-M" at the main screen, will place you in the modeling mode. Here, you will be able to organize items into model specific groups and start to build the data dictionaries required for the various models.

6.2. Summary of Commands

6.2.1. Keyboard Functions

It may be helpful to remember that the control sequence with the function keys will cause an associated form to appear immediately. For example, at the Collection screen, "F1" will place a list of activities on the screen while "Ctrl-F1" will place a blank activity form on the screen. During data entry, pressing "F1" will cause the addition of the specified activity to be placed in the database, while "Ctrl-F1" will place that activity in the database and bring up its form on the screen immediately.

Each function key has a classification associated with it. In order from F1 to F10, these are, activity, physical object, artifact, organization, system, process/scenario, event, link, descriptor, and statement. Four alphanumeric keys when used in conjunction with the control key will produce dialogue windows. Each key will classify the statement. These keys are "Ctrl-G," "Ctrl-B," "Ctrl-R," "Ctrl-P." Their respective statement classifications are general, business rule, relation, and problem.

The "Esc" key is the key to use to exit a form or screen. In all cases except during entry of text in a form, its meaning is just "exit". During entry of text in a form, its meaning is disregard this text modification and exit the text entry. You will remain in the current form and be allowed to attempt modification to that text again or any other normal form operation.

6.2.2. Main Screen Commands

- Ctrl-C - Go to the Collection Screen.
- Ctrl-A - Go to the Analysis Screen.
- Ctrl-M - Go to the Modeling Screen.
- ESC - Exit to operating system.

6.2.3. Collection Screen Commands

- Ctrl-S - Go to the Stage Description Form.
- Ctrl-G - Open a dialogue window to accept a general statement.
- Ctrl-B - Open a dialogue window to accept a business rule.
- Ctrl-R - Open a dialogue window to accept a relation statement.
- Ctrl-P - Open a dialogue window to accept a problem statement.

- F1 - Display a list of activities and select one activity to browse or modify its form.
- F2 - Display a list of physical objects and select one to browse or modify its form.
- F3 - Display a list of artifacts and select one to browse or modify its form.
- F4 - Display a list of organizations and select one to browse or modify its form.
- F5 - Display a list of systems and select one to browse or modify its form.
- F6 - Display a list of processes/scenarios and select one to browse or modify its form.
- F7 - Display a list of events and select one to browse or modify its form.
- F8 - Display a list of links and select one to browse or modify its form.
- F9 - Display a list of descriptors and select one to browse or modify its form.
- F10 - Display a list of statements.
- Ctrl-F1 - Display a blank form to enter an activity.
- Ctrl-F2 - Display a blank form to enter a physical object.
- Ctrl-F3 - Display a blank form to enter an artifact.
- Ctrl-F4 - Display a blank form to enter an organization.
- Ctrl-F5 - Display a blank form to enter a system.
- Ctrl-F6 - Display a blank form to enter a process/scenario.
- Ctrl-F7 - Display a blank form to enter an event.
- Ctrl-F8 - Display a blank form to enter a link.
- Ctrl-F9 - Display a blank form to enter a descriptor.
- ESC - Backup to main menu.

6.2.4. Analysis Screen Commands

Since this has not yet been implemented, exact key functions cannot be described at this time.

6.2.5. Modeling Screen Commands

See the MEDD users manual Appendix on of this report for the detailed commands for this mode.

6.2.6. Dialogue Window Commands

Once in a Dialogue window the user is provided with support for the entry and classification of text.

- Text entry
 - Alphanumeric keys - place their associated character at the cursor location. The current FCT supports word wrap. Note that since the concept of a dialogue entry is for the capture of simple declarative sentences the editor only supports the entry of 200 characters. A future version of this tool will support entry of the complex compound sentences typically required to capture conditional facts. However this mode is not supported in the current tool.
 - Enter - Accept the statement as is and return to the previous screen.
 - Esc - Accept the statement as is and return to the previous screen.
- Cursor Movement Commands
 - Home - Move cursor to beginning of current line.
 - End - Move cursor to end of current line.
 - Left - Move cursor one character to left.
 - Right - Move cursor one character to right.
 - Up - Move cursor to previous line.
 - Down - Move cursor to next line.
 - Ctrl-Left - Move cursor one word to left.
 - Ctrl-Right - Move cursor one word to right.
 - PgUp - Move cursor to top line.
 - PgDn - Move cursor to bottom line.
 - Ctrl-PgUp - Move cursor to beginning of statement.
 - Ctrl-PgDn - Move cursor to end of statement.

- Ins - Toggle edit mode between insert and overwrite.
- Del - Delete character above cursor.
- Backspace - Delete character to left of cursor.
- Text Marking Commands
 - Ctrl-B - Mark the beginning of a block of text.
 - Ctrl-E - Mark the end of a block of text.
 - Ctrl-C - Copy marked block to current cursor location.
 - Ctrl-D - Delete marked block.
 - Ctrl-E - Move marked block to current cursor location.
- Text Reference Classification Commands
 - F1 - Mark any of the thirty-five characters before the cursor that have not yet been classified as a block. If another F1 is pressed before any other key, the marked block will be reduced by one word. After hitting any other key, the marked block will be classified as an activity with a link back to the statement.
 - F2 - Same as F1 except the block will be classified as a physical object.
 - F3 - Same as F1 except the block will be classified as an artifact.
 - F4 - Same as F1 except the block will be classified as an organization.
 - F5 - Same as F1 except the block will be classified as a system.
 - F6 - Same as F1 except the block will be classified as a process/scenario.
 - F7 - Same as F1 except the block will be classified as an event.
 - F8 - Same as F1 except the block will be classified as a link.
 - F9 - Same as F1 except the block will be classified as a descriptor.

The following commands support the classification of a text block and immediate description of that reference. Upon completion of the associated description capture form the user will be returned to the point in the dialogue text capture form at which the command was executed.

- Ctrl-F1 - Classify any of the thirty-five character before the cursor that have not yet been classified as an activity and display that activity's form for data entry.
- Ctrl-F2 - Same as Ctrl-F1 except the block will be classified as a physical object.
- Ctrl-F3 - Same as Ctrl-F1 except the block will be classified as an artifact.
- Ctrl-F4 - Same as Ctrl-F1 except the block will be classified as an organization.
- Ctrl-F5 - Same as Ctrl-F1 except the block will be classified as a system.
- Ctrl-F6 - Same as Ctrl-F1 except the block will be classified as a process/scenario.
- Ctrl-F7 - Same as Ctrl-F1 except the block will be classified as an event.
- Ctrl-F8 - Same as Ctrl-F1 except the block will be classified as a link.
- Ctrl-F9 - Same as Ctrl-F1 except the block will be classified as a descriptor.

6.2.7. Item Description Form Commands

An Item Form is any of ten different forms used by this tool to acquire more detailed information about a specific item or area. These include the activity, physical object, artifact, organization, system, process/scenario, event, link, descriptor forms and the stage form. There are two modes for each form; (1) the data entry mode where the user is typing in characters, and (2) the entry prompt movement mode in which different areas may be highlighted (selected for editing).

• Form Data Entry Mode Commands

- alphanumeric keys - place their associated character at the cursor location.
- Home - Move cursor to beginning of the entry text.
- End - Move cursor to end of the entry text.
- Left - Move cursor one character to left. Upon the second consecutive left arrow key depression with the cursor at the end of the allocated length of the entry, the entry will be accepted as is and the entry prompt will be moved to the next entry.
- Right - Move cursor one character to right. Upon the second consecutive right arrow key depression with the cursor at the beginning of the entry, the

entry will be accepted as is and the entry prompt will be moved to the previous entry.

- Ins - Toggle the edit mode between insert and overwrite.
- Del - Delete the character above the cursor.
- Backspace - Delete the character to the left of the cursor.
- Enter - Accept the entry as is.
- Esc - Do not accept entry. Leave entry as it was before editing was begun.
- Entry Prompt Movement in Data Entry Forms
 - Home - Move entry prompt to beginning of the form.
 - End - Move entry prompt to end of the form.
 - Left - Move entry prompt one entry to left.
 - Tab - Move entry prompt one entry to left.
 - Right - Move entry prompt one entry to right.
 - Sft-Tab - Move entry prompt one entry to right.
 - Up - Move entry prompt to entry on previous line.
 - Down - Move entry prompt to entry on next line.
 - PgUp - Move entry prompt to entry one page (twenty-five lines) before the current entry.
 - PgDn - Move entry prompt to entry one page (twenty-five lines) after the current entry.
- Item classification within Data Entry Forms
 - F1 - Mark the current entry as an activity with a link back to the current form.
 - F2 - Same as F1 except the block will be classified as a physical object.
 - F3 - Same as F1 except the block will be classified as an artifact.
 - F4 - Same as F1 except the block will be classified as an organization.

- F5 - Same as F1 except the block will be classified as a system.
- F6 - Same as F1 except the block will be classified as a process/scenario.
- F7 - Same as F1 except the block will be classified as an event.
- F8 - Same as F1 except the block will be classified as a link.
- F9 - Same as F1 except the block will be classified as a discriptor.
- Ctrl-F1 - Classify the entry as an activity and display that activity's form for data entry. Upon completion of data entry for this activity, return to this form.
- Ctrl-F2 - Same as Ctrl-F1 except the block will be classified as a physical object.
- Ctrl-F3 - Same as Ctrl-F1 except the block will be classified as an artifact.
- Ctrl-F4 - Same as Ctrl-F1 except the block will be classified as an organization.
- Ctrl-F5 - Same as Ctrl-F1 except the block will be classified as a system.
- Ctrl-F6 - Same as Ctrl-F1 except the block will be classified as a process/scenario.
- Ctrl-F7 - Same as Ctrl-F1 except the block will be classified as an event.
- Ctrl-F8 - Same as Ctrl-F1 except the block will be classified as a link.
- Ctrl-F9 - Same as Ctrl-F1 except the block will be classified as a descriptor.
- Exiting Data Entry Forms
 - Enter - No meaning.
 - Esc - Accept the form as is and return to previous screen.

Appendix A

Installation and Hardware and Software Requirements fo

1.1. IDEF0 and IDEF1 Hardware and Software Requirements

The MDSE requires specialized hardware to be executed. A minimal system to run the MDSE would be a Symoblics 3600 series computer with the following configuration:

1. Two million words of memory.
2. One hundred and forty million bytes of on-line disk storage.
3. A cartridge tape drive.

A later model Symbolics (3620 or 3650) and systems with more than two megawords of memory would increase performance. Additional disk storage would be useful when a large number of IDEF models must be available for modelers. The tape drive is necessary to install the MDSE onto the local machine.

Software requirements for the MDSE are simply that Genera version 7.1 be installed on the host machine. Genera is the Lisp-based operating system for Symbolics computers. Version 7.1 is the only version currently compatible with the MDSE (at the time of this writing, the MDSE is not compatible with Genera version 7.2). Consult Symbolics documentation for installation of Genera on the host machine.

1.2. IDEF0 and IDEF1 Installation Procedures

IDEF0 and IDEF1 Installation Procedure

To install the IDEF0 and IDEF1 modeling tools, approximately 800 blocks of disk space must be available on the host where the tools will be located. The following steps must be performed in order, with the installer optionally substituting the name of the host on which the modeling tools will be installed where "local" appears below.

1. Create the following top-level directories on the destination host:

```
LOCAL:>MODEL-BUILDER>  
LOCAL:>METAMODELER>  
LOCAL:>IDEF0>  
LOCAL:>IDEF1>
```

2. Evaluate the following Lisp forms either by typing them into the Lisp Listener or entering them into a ZMACS editor buffer (and saving it into the file named "SYS:SITE;MODEL-BUILDER.TRANSLATIONS" for future use):

```
(fs:set-logical-pathname-host "MODEL-BUILDER"
 :physical-host "LOCAL"
 :translations '(("MODEL-BUILDER;**;" "LOCAL:>MODEL-BUILDER>**>")
                  ("**;" "LOCAL:>MODEL-BUILDER>**>"))))
```

```
(fs:set-logical-pathname-host "METAMODELER"
 :physical-host "LOCAL"
 :translations '(("METAMODELER;**;" "LOCAL:>METAMODELER>**>")
                  ("**;" "LOCAL:>METAMODELER>**>"))))
```

```
(fs:set-logical-pathname-host "IDEF0"
 :physical-host "LOCAL"
 :translations '(("IDEF0;**;" "LOCAL:>IDEF0>**>")
                  ("**;" "LOCAL:>IDEF0>**>"))))
```

```
(fs:set-logical-pathname-host "IDEF1"
 :physical-host "LOCAL"
 :translations '(("IDEF1;**;" "LOCAL:>IDEF1>**>")
                  ("**;" "LOCAL:>IDEF1>**>"))))
```

3. Place the distribution tape containing the modeling tools into the tape cartridge drive.
4. Type "Restore Distribution" into the Lisp Listener to load the tools onto the local host.

Loading of the modeling tools should take about ten minutes to complete. If the system prompts the installer to confirm any subdirectory creation, answer 'yes' to allow the restoration to continue.

Appendix B

Installation and Hardware and Software Requirements fo

2.1. Setting up Directories

The MEDD files are contained on one high density diskette. To be used they must be copied to a hard drive. Use the following steps to copy the files:

1. From the root directory on the hard drive create a sub-directory named QPRO. At the DOS prompt, type the following:
 - **MD QPRO**
2. From the root directory on the hard drive create a sub-directory named QPMETH. At the DOS prompt, type the following:
 - **MD QPMETH**
3. From the root directory on the hard drive create a sub-directory named MOUSE1. At the DOS prompt, type the following:
 - **MD MOUSE1**
4. Put the MEDD disk in drive A.
5. Change the current directory from root to QPRO. At the DOS prompt, type the following:
 - **CD \ QPRO**
6. Copy the files from the QPRO sub-directory on the MEDD disk in drive A to the QPRO sub-directory on the hard drive. At the DOS prompt, type the following:
 - **COPY A:\ QPRO\ *.***
7. Change the current directory from QPRO to QPMETH. At the DOS prompt, type the following:
 - **CD \ QPMETH.**
8. Copy the files from the QPMETH sub-directory on the MEDD disk in drive A to the QPMETH sub-directory on the hard drive. At the DOS prompt, type the following:

- **COPY A:\ QPMETH *.***

9. Change the current directory from QPMETH to MOUSE1. At the DOS prompt, type the following:

- **CD \ MOUSE1**

10. Copy the files from the MOUSE1 sub-directory on the MEDD disk in drive A to the MOUSE1 sub-directory on the hard drive. At the DOS prompt, type the following:

- **COPY A:\ MOUSE1 *.***

11. Change the current directory from MOUSE1 to root. At the DOS prompt, type the following:

- **CD **

12. Copy all remaining files from the MEDD disk in drive A to root directory on hard drive. At the DOS prompt type the following:

- **COPY A: *.***

2.2. Modifying AUTOEXEC.BAT and CONFIG.SYS

In order for the MEDD to operate properly, the DOS files AUTOEXEC.BAT AND CONFIG.SYS must be modified as follows:

1. Modify the AUTOEXEC.BAT file. Include the following statements:

- **\MOUSE1 \ MOUSE \ MOUSE1 \ CPANEL**

2. Modify the CONFIG.SYS file. Include the following statements:

- **BUFFERS=25 FILES=20**

3. Reboot System -- The system needs to know what changes have been made to AUTOEXEC.BAT and CONFIG.SYS before it can invoke the MEDD. It is for this reason that the system must be rebooted.

Reboot system (Ctrl-Alt-Del).

2.3. Invoking MEDD Via Q-PRO4

The following steps are required in order to invoke the MEDD after it has been installed:

1. Start the Q-PRO4 Format Building (FB) Editor via the QP batch file. At the DOS prompt (root directory), type:

- QP

The main menu of the Q-PRO4 Format Builder should now be displayed.

2. Select N to enter the name of the desired format file. When prompted for the format file name, type the following:

- \ QPMETH\ MM.QNE <ENTER>

Select R to run the Q-PRO4 runtime interpreter (QNE). The MEDD application should be invoked.

The MEDD application files on the MEDD disk are interpretive. This means that the Q-PRO4 runtime interpreter is required to operate the MEDD application. The interpreter works only with files that have the extension .qne (MEDD application files), .fid (file item description files), .dat (database files) and .\$\$\$ (sequential files). The Q-PRO4 interpreter should be installed on a hard drive in a QPRO directory [use the DOS MAKE DIRECTORY (MD) command]. The MEDD files should be installed on the same drive in a QPMETH directory [use the DOS MAKE DIRECTORY (MD) command]. The Q-PRO4 interpreter executes from within the QPRO directory and utilizes the MEDD application files which reside in the QPMETH directory.

2.4. Hardware and Software Requirements

2.4.1. Computers

This application was designed to operate on the IBM PC family of computers (i.e., the PC/XT, PC AT and PS/2) and computers that are 100% compatible with IBM (including laptop portables with hard drives). Note: 8086/8088 and 80286 based microsystems have been tested.

2.4.2. Displays

This application was designed to operate with the IBM Monochrome Display, IBM Color Display, IBM Enhanced Color Display, IBM Personal System/2 Displays, and Displays that are 100% compatible with one of the displays mentioned herein.

2.4.3. Display Adapters

This application was designed to operate with the following display adapters:

1. IBM Monochrome Adapter,
2. IBM Color/Graphics Adapter,
3. IBM Enhanced Graphics Adapter,
4. IBM Personal System/2 Display Adapter,
5. Hercules Graphics Card,

and display adapters that are 100% compatible with one of the display adapters mentioned herein.

2.5. Other Requirements

1. Pointing Device -- Microsoft Mouse (Serial or
 2. Bus Interface, depending on system configuration)
 3. Peripherals -- printer (dot matrix style)
 4. Microprocessor Memory Requirements -- 512K Bytes or higher
 5. Operating System Requirements -- Microsoft DOS Version 2.0 or higher
 6. Software Requirements
 - a. Q-PRO4 Forth Generation Applications Development System For Micro-computers.
 - b. Microsoft Mouse Software (MOUSE.COM and CPANEL.COM).
- Disk Formats-- 5 1/4in. and 3 1/2in.

2.6. Distribution and Support of the MEDD

The MEDD is a prototype system, developed to bridge the gap between the collection of raw facts, and the structural model development. However the MEDD is usable as a stand alone tool. The product delivered under this effort is written primarily in the Q-PRO4 language. To execute this software requires the Q-PRO4 system and the Microsoft Mouse software. As of the preparation of this users manual the Knowledge Based Engineering Enterprise (KBEE) company as distributors

of the Q-PRO4 product have agreed to produce executable versions of the MEDD for those interested in stand-alone operation for the cost of the appropriate run time license, handling, and shipping charges. The Q-PRO4 system supports the generation of a "C" language version of any software produced with the Q-PRO4 language. As such a version would contain components proprietary to Q-PRO Inc. it was not possible to deliver such a version to the sponsor for distribution. Again, for the cost of the appropriate run time licenses, handling and shipping charges KBEE has agreed to provide interested users with this version of the product.

Appendix C

Installation and Hardware and Software Requirements fo

Due to the heavy disk usage by this tool, it is recommended that a hard disk be used to maintain a suitable speed of operation. It is also recommended that a new directory be made in which to place all of the necessary files.

3.1. Floppy Disk Installation

The following steps are required to install the FCT on a new diskette for use in systems without hard drives.

1. Insert a blank formatted disk into drive B.
2. Insert the DCT disk into drive A.
3. At the DOS prompt type

- `copy a:dct.exe b:`

The first time that the DCT is run, it will create all of the necessary database files. It is always a good idea to periodically make backups of the system and the database files on diskettes.

To run the DCT program make the drive with the disk containing the DCT the current drive. At this time the DCT disk is in drive B. Type the following:

- `b:`
- `dct`

3.2. Hard Disk Installation

It is assumed that the hard disk is drive C. If this is not so, replace all occurrences of `c:` with the drive letter of the drive being used followed by a colon.

1. To create a directory, use the DOS make directory command. Choose a name that is meaningful and less than eight characters long. The name of the directory in this example is "DCT". First make drive C the current drive. Then create the directory. And then make that directory the current directory. To do this, type the following:

- `c:\`

- md \dct
 - cd \dct
2. Now place the DCT disk in drive A and copy the DCT file to drive C.
 - copy a:dct.exe
 3. The first time that the DCT is run, it will create all of the necessary database files. To run the DCT program,type:
 - dct

3.3. Technical Information

The C language was chosen for this tool because of its power, speed, and widespread use in industry. Development was performed using Microsoft's C 5.0 Compiler and supports all of the standard C functions. Novell's Btrieve was used for the database management.

The tool has been written in a modular form to allow modifications to one area and not affect the entire program. For example, the function calls that affect the databases may be modified to use any database package other than the one currently in use. After properly modifying and compiling this one module, the entire tool may be linked again without fear of disturbing the the rest of the program.

The commented source code for all of the routines except the database package is available and included with the executable program.

Appendix D

Introduction to Symbolics

The following section is supplemental information concerning the Symbolics and its operating environment. This information is not necessary reading for operation of the IDEF modeling tools, it is only intended to be used by those who wish to further understand the Symbolics and its operating environment. The ensuing section provides information concerning the Keyboard, Conventions and Definitions which are important to the operation of the IDEF Modeling Tool.

If you have never used the IDEF modeling tools before, please take the time to read the Introduction to the IDEF Modeling Concept (section 1) and the Introduction to the Symbolics (section 3). These sections introduce you to the overall structure of the IDEF modeling tools, helpful keys to use, how to login and logout, and much more.

4.1. Conventions and Definitions

The following conventions are common phrases and their definitions that are used throughout the "Introduction To Using The Symbolics." The following notation, <>, " ", [] and bold letters are used so that you can distinguish between a control sequence command, user typed input and window names, current values of a parameter, and menu selections, respectively.

The following are examples of this notation:

<CONTROL-ABORT> - The parentheses indicate that this is a control sequence command that the user must type if function is needed. The dash indicates that the keys must be pressed simultaneously.

" " "YES" or "Command Window" - The quotes are used to indicate that there is user input needed or that window name is mentioned.

"bold" used for **CREATE** or **SAVE** - The bold face is used to indicate that an item is a selectable menu item. Also used to indicate feedback by the machine such as the : prompt or questions which are asked by the system that require typed user input.

Press **SELECT L** - Press and then release the <SELECT> button and then press the L.

Press **CONTROL-ABORT** - Press and hold the **CONTROL** button while simultaneously pressing **ABORT**. Then immediately release both keys.

Mouse Middle Twice - Press the middle button on the mouse twice in rapid succession. Refer to Figure 15

Click Left - Press the left button of the mouse.

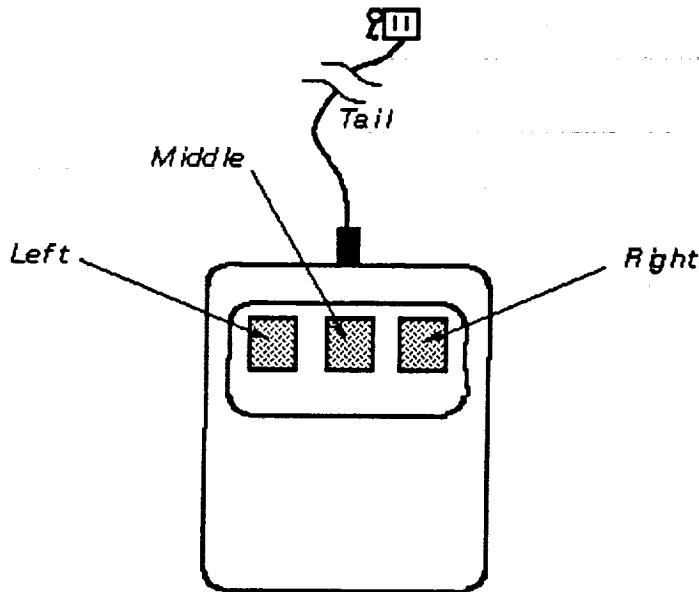


Figure 15. Symbolics Workstation Mouse

Mouse Status Line - The reverse video line at the bottom of the screen explains the effects of pressing the L, M, R mouse buttons. (see Figure 16) For example, you can press the left mouse button twice for "L2" or you can press the shift key and click left with the mouse. The button clicks are coded as follows:

- L = Press the left button
- L2 = Press the left button twice in quick sequence
- M = Press the middle button
- R = Press the right button

System Status Line - Line below "Mouse Status Line" which shows the time, date, and who is logged in. Refer to Figure 16

Modeler Status Line - This line is above the Model Builder Display Pane. It gives the status on the model name, the view, and whether or not the pane is selected (active).

```

Mouse-II: Menu.
To see other commands, press Shift, Control, Meta-Shift, or Super.
[Mon 28 Dec 9145:28]  KJones  CL USER:  User Input
date  time  user  name  syntax  package  name  process  state
  
```

Figure 16. Symbolics Workstation System Status Line

Command Window - The command window is the lower left window in which you issue all of your IDEF1 and IDEF0 commands. You also will be able to get help in this window by pressing the HELP key. Also, any error messages which might occur will be received in this window.

Text Display - The Text Display window is the model builder window displayed on the lower right of the model builder 1-display window. It displays various information.

Multiple Panes or Windows - Multiple Panes or windows can be created by using the command Select Configuration. It will allow you to create up to three display panes at one time.

4.1.1. Screen Configuration

Print The Screen - Press FUNCTION Q and wait until you see a message below the mouse status line (dark bar at the bottom of the screen in the lower right hand corner).

4.2. Keys to Become Familiar with

The following section provides the user with a list of all keys important to the operation of the IDEF modeling tools. Please take the time to review this list, unless you are already familiar with the Symbolics machine.

The following is a list of all the keys that are important to the operation of the IDEF Modeling Tool. On the left you will find the name of the key(s) and on the right, a description of their function and how to operate the keys (i.e. meaning if the key(s) operate by pushing two simultaneously or if there is a certain ordering of two or three keys). Refer to Figure 17

Help - Pressing the HELP key will provide online documentation designed to assist the user in the current context. Also, by pressing any of the function

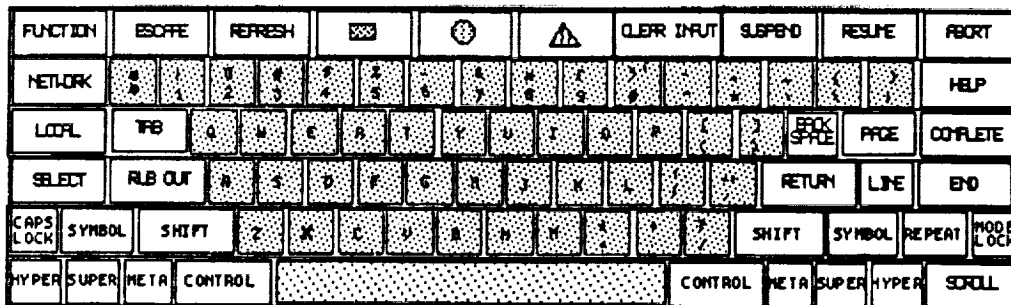


Figure 17. Symbolics Workstation Keyboard

keys (below) followed by the **HELP** key will provide on-line documentation of commands using that key. (i.e. **SELECT HELP**)

Function - Function allows you to perform operations on windows and processes. Press **FUNCTION HELP** to see a listing of all the actions that can be taken by pressing **FUNCTION** key combinations.

Select - The system programs on the Lisp machine are called processes. The Lisp Listener and the editor are two of these processes. Frequently, you will want to go from one of these processes to another. There are several ways to switch between processes by using the **select** key followed by another key input. If you press **SELECT HELP**, then you will get a listing of all the machine's processes which are accessible through the **select** key.

Abort - Aborts the operation currently in progress. It returns to the command level in such programs as the Lisp Listener, Editor, File System Editor , and so forth. It takes effect when the program reads it, rather than immediately.

Control - Control is considered a modifier key, it modifies the meaning of other keys. To see what commands the **CONTROL** key is used for, press **CONTROL-HELP**. It is always pressed simultaneously with another key, not sequentially like the **FUNCTION** or **SELECT** key.

Meta - Meta is also considered a modifier key, it modifies the meaning of other keys. It is often used with the **CONTROL** key, both are held down simultaneously. For example, **C-M-RUBOUT** is for deleting the previous LISP expression.

Select L - Pressing **SELECT L** will take you to the Lisp Listener, for example, if you are in the Editor and wish to get back to the Lisp Listener.

- Rubout** - Rubout will delete the previous input by one character or the last key command that was issued.
- Suspend** - Suspend will temporarily suspend or halt the action of the program and place you in an environment like the LISP Listener. You can resume your work by pressing RESUME.
- Control-Abort** - Control-Abort is like ABORT, except it is more serious.
- Meta-Abort** - Meta-Abort causes the process to return to topmost command loop, aborting all computation and other command loops which might be on the stack. It takes effect when read, rather than immediately.
- Function C** - By pressing FUNCTION and then releasing, then pressing C, this will toggle you back and forth between states of black-on-white and white-on-black screen representation, sometimes referred to as reverse video or inverse video.
- Local B** - By pressing LOCAL and B simultaneously this will brighten the screen. Sometimes when you begin your session on the Symbolics, if you should sit down and the screen appears to be off, before you begin turning switches try LOCAL-B first because most likely someone has just dimmed the screen.
- Local D** - By pressing LOCAL and D simultaneously, this will darken the screen. This allows you to turn down the brightness and also completely darken the screen when not using the Symbolics to help conserve the screen.
- Clear Input** - Clear Input erases all characters typed since the last command prompt.
- Refresh** - Refresh clears the window, sometimes redrawing the contents. For instance, in a Lisp Listener, pressing REFRESH will clear the window.
- Repeat** - Repeat causes a key being pressed at the same time to repeat. Normally, no matter how long you hold a key down, only one of that character is sent to the machine. (There are ways to make the keys repeat automatically.)
- Symbol** - The character set contains more characters than there are keyboard keys, even when they are combined with the modifier keys. SYMBOL allows you to access special characters. Type SYMBOL-HELP for a listing of all special characters.

4.3. Logging In and Out

4.3.1. Logging In

If the machine is on, but the display is blank, first try making the screen brighter. This is done by pressing and holding down LOCAL-B until the screen is the desired brightness. If you overshoot, holding down LOCAL -D dims the screen. If this does not do anything, the power switch for the display is on the back. Check to see if it is turned on. The machine should now be ready for you to login.

Logging in is an initial procedure that is necessary in order to run the IDEF Modeling Tools. In order to login, you must have a user ID created. Most likely the person who is in charge of system maintenance will be the one who will create your user ID (the procedure for creating your user id will be given below in one of the following paragraphs).

Logging in on a Lisp Machine records your name for file operations and loads a file (your lisp-init file) from your home directory. On the Lisp Machine, your login-name can be any string (as long as the string does not conflict with any login-name which is already present). It is most common to use first or last names, initials or nicknames as your login-name.

NOTE: A lisp-init file can contain Lisp code which allows you to customize your Lisp environment. You can login without a lisp-init file but the machine notifies you that the file cannot be found.

The following is a step-by-step procedure of how you are to login once your user ID is created. The following format of steps should allow you to see what your input should be and what the computer output should look like.

NOTE: Always be sure to check the status line at the bottom of the screen, to make sure no user name is present. If you should see that someone is logged in, you should check with that person before logging them out because important files could be lost if not saved properly.

1. Type "Login" <SPACEBAR> You will see: Command: Login (user name)

Once you type "Login" and hit the SPACEBAR, the computer will ask for your "(user name)" which you proceed with to type on the same line as "Login". Let us assume the user name is John.

2. Complete the login command by typing your user name, John. Once you have typed in your user name, hit <RETURN> and the computer will start to load your files in your directory.
3. You will see: Command: Loading [machine name]: John>lisp-init.lisp into package USER (really COMMON-LISP-USER). John should be typed on the same line as Login.
4. Once the machine is finished loading your directory it will come back with the COMMAND: prompt. You are then ready to begin your session.

4.3.2. Logging Out

When you are finished with the machine, you should save, in files, any work you want to keep, and then log out. Logging out informs the machine that you're leaving; it will query you about any editor (Zmacs) buffers which have not been saved in case you have forgotten.

The following is a step-by-step procedure of how you are to logout once you are finished with the IDEF Modeling Tool.

1. You must be in the Lisp Listener in order to log out and you can do this by entering the command `SELECT L`. If you are already in the Lisp Listener, the screen will flash.

NOTE: If you see the line ****MORE**** at the bottom of your screen, press the **SPACEBAR**.

2. Once you are in the Lisp Listener, you are then ready to logout. Type **"Logout"** **RETURN**.

4.4. Powering up the Machine

If the machine is powered down

1. First turn on the display. The switch for the display is on the rear at the right.
2. Press in the green power button on the front of the base unit. If by pressing the button it pops out, then the machine was shut off by software control (the preferred method) and no one pressed the power button afterwards. Merely press the button again.
3. When it asks you to type **"Hello,"** do so. If you get the message **"Drive 0 not ready -retrying... press any key to abort"**, the disk drive is just spinning up. If it lasts more than 60 seconds, the disk did not synchronize properly -- Turn off the machine, wait 90 seconds, turn the machine back on, and type **"Hello"** when requested.
4. Next you must Cold-boot the machine.

4.5. Cold-Booting the Machine

Cold booting can be done at the end of a session or at the beginning of a session. When you first sit down at a machine, you want to begin work in a clean, unaltered computing environment. Cold-booting the machine will provide an environment, or world, which is clean. You can think of your Lisp world as a blackboard. When someone uses it, it gets filled up with what they have done. If you want to use it, you wipe it clean (cold-boot) so that you begin work on a clean slate.

You can tell if the machine has been cold-booted by looking at the lower right-hand corner of the screen. If you do not see the words, **[machine] is cold-booted"**, then the world has been modified by someone's use since the machine was last

cold-booted. Remember to explicitly save all of your work in files, because only information in files, which are stored on disk, survive cold-booting.

Always remember to check the status line to see if there is a user name present. If so, it is a good idea to check with the previous user of the machine because that user may need to save their work.

The following is a step-by-step procedure for cold-booting the machine:

1. You should be in the Lisp Listener to cold boot the machine so begin by typing **SELECT L**. You should see the "Command:" prompt.
2. This is the prompt for the Command Processor. Look at the status line. If there is someone currently logged in to the machine, type: "**Logout**" **RETURN** to log them out. If you see ****MORE**** at the bottom left of the screen, press the **SPACEBAR**.
3. If you log someone out, you may be asked about saving buffers. If so, you should ask the previous user of the machine if any of their work should be saved, and then enter "**Y**" or "**N**".
4. After the Logout command completes, you see the message: Logged out When the **COMMAND:** prompt appears again, type: "**Halt**" **SPACEBAR** "**Machine**" **RETURN**.
5. Look at the upper left corner of the screen. You will see the following: Lisp stopped itself FEP command: This means that you are now addressing the Front End Processor (FEP), and have to use the FEP Boot command to execute the commands in the boot file, which is a file of FEP commands.
6. To execute the commands in the boot file, you should type: "**boot**" **SPACEBAR** (default is **Fep:boot.boot**) **RETURN**. You will then see the output: **Fep Command: Clear Machine Files..... etc...**

Several messages appear on the screen. You will see the microcode being loaded, the world being loaded, paging files being loaded, etc. Finally, you will have a screen which says Lisp Listener 1 in the lower left-hand corner, and (your machine's name, Laser) is cold-booted on the bottom line.

Once you are in the Lisp Listener 1 window, you are now being prompted with:

Please login.

COMMAND:

4.6. Creating a User ID

In the section "Logging In", you were told step-by-step how to login to the Symbolics. As you probably noticed, the machine asked for a user name. This user name is actually your user ID that must be created prior to beginning your first session on the IDEF Modeling Tool. The Symbolics recognizes your user id in the namespace database. This next section will explain a few things about the namespace object database and show you how to add yourself to that database.

The association of your name with a particular file-server is made by the database known as the namespace. The namespace contains information about users, hosts, printers, and other objects. In particular, the information about users includes which home host is associated with which login-name. Your login-name is the name by which the machine knows you. Your home host is the machine on which you keep your files. The machine looks for your `lisp-init` file in the directory `[file-server:login-name]`. This directory is your home directory. If you do not appear in the database, you are invited to add yourself when you first log in. You only have to do this once.

To add yourself to the namespace, you should use the Namespace Editor. If you have difficulty logging in because the machine does not recognize your login-name, you are asked if you wish to add yourself to the namespace database. If you say "Yes", you are put into the Namespace Editor automatically, and you are editing your user record.

The pieces of information about a user are known as attributes. You must enter four of these attributes to add yourself to the namespace. The four required attributes are marked with asterisks (*) after their names. These attributes as they appear on-line are as follows:

- **Lisp Name*:**
- **Personal Name*:**
- **Home Host*:**
- **Mail Address*: Pair:**

You can also access the Namespace Editor from the System Menu. If you select it this way, you are not initially editing a record. We will only show you how to add yourself to the namespace when you can't log in. You should not experiment with adding other objects to the namespace unless you are sure you know what you are doing.

The following is a step-by-step procedure on how to add yourself to the namespace when you have a problem logging in.

1. The first thing to do is check to see whether you have typed your login-name properly. If not, press `<ABORT>` and try to log in again. It is a good idea to ask your site administrator to help you login if you are having problems. If

you do not have a site administrator available, follow the steps below. You should only need to do this once.

2. When you try to log in, (suppose we are trying to login with the user name John) if you are not part of the user database or if you have mistyped your login-name, You will see the following:

```
The user named 'john' was not found.:  
Do you want to log in as John  
on some specific host? (Y, N, or R)
```

3. Your choices are (y) yes, (n) no or (r) retry.
4. If you feel the machine has made an error, you can do a (R) retry. Suppose we do a retry, we would see:

```
Retry user object lookup.
```

5. In case it turns out that you do not have a user id and that you must create one, you would make the selection of (Y) yes. You would then see:

```
Host to login to:
```

6. You should type the name of your file-server, the machine on which your files are stored. All the hosts that are known by the namespace database have names.

7. You will then see:

```
No init file: The directory JOHN does not exist.  
For [machine name]: john lisp-init.bin  
Do you wish to add john to the user database? (Y or N)
```

8. You should type the letter Y for yes. You are now presented with a new window. This is the Namespace Editor. Here you are told by the namespace database, exactly how to add a user object for yourself to the namespace database.

9. You should see a screen which has two windows, a top window and bottom window. The cursor will appear in the bottom window with a message that says:

```
Click on an attribute entry to replace (L), delete (M) or  
edit (R) it.
```

10. The Personal Name attribute must be added. To do this, move the mouse until the mouse cursor is close to the word Token following Personal Name*:. The cursor changes into a hollow rectangle surrounding Token. Carefully click left. The machine prompts you in the bottom window for the personal

name of the user. For this example it will be John. You would type in your name, followed by RETURN. Your name appears in the top window, next to the attribute you had previously clicked on.

NOTE: It is important that we caution you to not change any other attributes unless you are an experienced user and know what you are doing. The Personal Name*: attribute is the only attribute you need to change at this time.

11. Now that you have completed the previous step, move the mouse cursor to the Save option which is located in between the top and bottom windows of this menu. You will notice that there are eleven options to choose from but the only ones important to you at this time are Save and Quit. Click left on Save. This saves the user object in the database.
12. After the message, "the user named Site|Login-name has been saved" appears, move the mouse to the Quit option. Click left on Quit. The window disappears, leaving you in Dynamic Lisp Listener 1. You have now added yourself to the namespace, and you can log in normally.

4.7. Getting Around in the Symbolics World

The Symbolics is an unusual machine in that it will keep numerous functions, or working environments, alive at one time, and you may move from one to the other quite easily. Thus if you are working with IDEF0, and you decide you want to look at the files in your directory, for instance, you can abruptly switch to the file environment without losing the IDEF0 environment. It will be waiting suspended exactly where you left it. An exception is if IDEF0, or any program, is in execution (not in a halted state). In this case, you can still move to a different environment, but execution will continue in your absence. In general, you just press the <SELECT> key followed by a one letter code for the environment you want to go to. For instance, <SELECT F> for the file system.

If you wish to list the functions that the machine has at any time in your session, it is possible by pressing <SELECT> then by pressing <HELP>. A "select help" menu will come up on-line with all the functions that are currently available on that machine.

Once you have entered an activity, you never exit it until the machine has been re-booted. So if you are in one activity and then you select another, you really have not exited the previous activity, you merely just selected a second activity. Therefore, when you return to a activity that you selected previously but since the time that you re-booted, you will find that you are exactly where you left off.

4.8. Machine Activities

The following is a list of additional activities for the Symbolics. Those environments can be entered by simply pressing **SELECT**, and then the specific application you are wanting to use. The Symbolics that you are using might have more or less additional applications, it just depends on the particular machine that you are presently using.

<SELECT 3> - This selection will take you into the IDEF model builder tool once the files have been loaded.

<SELECT-L> Lisp Listener (this is also the 7.1 operating system).

<SELECT-E> Text Editor

<SELECT-D> Document Examiner (the complete manual set is in electronic form for Symbolics 7.1 Lisp, the text editor, etc).

Getting Help

Remember that if you should ever need help on any key, all you have to do is press the key in question + the **HELP** key.

Index

Model Application / Analysis Support 15
Model Object Save / Hardcopy 12

A

A

A

Machine Swap
Creating a Model, Activity 3
Create Activity command 23, 50
Delete Activity command 50
Move Activity command 50
Rename Activity command 50
Show Activity Decomposition command 53
Add Attributes to Key Class command 32, 55
Add Record 68
Add Record(s) 68
Insert After 68
All Displayed Instances 68
Delete All Projects In Data Dictionary System command 61
Alphabetically 68
Model Application / Analysis Support 15
Model Application / Analysis Support 15
application tutorial 65
Approvers 66
Architectural Issues 15
MEDD Architecture 65
Getting Around in the Symbolics World 103
Hints and Assumptions of IDEF0 30
Attribute Class 14
Create Attribute Class command 55
Delete Attribute Class command 55
Inspect Attribute Class command 55
Attribute Class Cross Reference 66
Attribute Class Definition 66
Edit Attribute Class Description command 55
Edit Inherited Attribute Class Description command 55
Attribute Classes 8
Edit Attribute Class in Key Class Description command 55
Attribute Class Migration Index 66
Attribute Class Pool 10, 15, 32, 66
Create Item for Attribute Class Pool command 32, 61
Interface Attributes 66
Remove Attributes From Key Class command 55
Add Attributes to Key Class command 32, 55
Modifying AUTOEXC.BAT and CONFIG.SYS 87
Auto Layout 12
Automatic Model Rule Enforcement 13
Auto Router 12

B

	B+ Tree File System	65
	Back Entity Class	32
Form Menu	Bar	68
Main Menu	Bar	68
	Keys to Become Familiar with	95
	Insert	Before 68
Loading the Model	Builder	20
General Model	Builder Commands	47
	Model	Builder Development Window 15
	Model	Builder Status Line 18
IDEF Model	Builder Status Line and Mouse Status Line	18
IDEF Model	Builder Windows	15

B

C

C

Link	Cardinality 32
	Change Context command 6, 52
	Change Name of Current Project Data Dictionary command 61
Wall	Chart Production 13
	Check command 55
	Check Decomposition command 50
Attribute	Class 14
Back Entity	Class 32
Entity	Class 14
Front Entity	Class 32
Relation	Class 14
Add Attributes to Key	Class command 32, 55
Create Attribute	Class command 55
Create Entity	Class command 55
Create Key	Class command 55
Create Link	Class command 32, 55
Delete Attribute	Class command 55
Delete Entity	Class command 55
Delete Key	Class command 55
Delete Link	Class command 55
Inspect Attribute	Class command 55
Move Entity	Class command 55
Move Link	Class command 55
Remove Attributes From Key	Class command 55
Attribute	Class Cross Reference 66
Attribute	Class Definition 66
Entity	Class Definition 66
Relation	Class Definition 66
Edit Attribute	Class Description command 55
Edit Attribute Class in Key	Class Description command 55
Edit Entity	Class Description command 55
Edit Inherited Attribute	Class Description command 55
Edit Key	Class Description command 55
Edit Link	Class Description command 55
Information Model Entity	Class Elements 66
Attribute	Classes 8
Entity	Classes 8
Link	Classes 32
Relation	Classes 8
Merge Entity	Classes command 55
Edit Attribute	Class In Key Class Description command 55
Attribute	Class Migration Index 66

Attribute	Class Pool 10, 15, 32, 66
Entity	Class Pool 10, 15, 32, 66
Create Item for Attribute	Class Pool command 32, 61
Create Item for Entity	Class Pool command 32, 61
Entity	Class Synonyms 66
Key	Class to Inherit 32
	Clear Command History command 50
	Clear Data 68
	Clear Output History command 50
	Cold-Booting the Machine 99
Fact	Collection Tool 73
Micro-Based Fact	Collection Tool User's Manual 73
Add Attributes to Key Class	command 32, 55
Change Context	command 6, 52
Change Name of Current Project Data Dictionary	command 61
Check	command 55
Check Decomposition	command 50
Clear Command History	command 50
Clear Output History	command 50
Concept Editor	command 23, 50
Copy Model Portion	command 55
Create Activity	command 23, 50
Create Attribute Class	command 55
Create Concept	command 23, 50, 52
Create Entity Class	command 55
Create Item for Attribute Class Pool	command 32, 61
Create Item for Entity Class Pool	command 32, 61
Create Key Class	command 55
Create Link Class	command 32, 55
Create Model	command 23, 47, 55
Create New Model Current Project Data Dictionary	command 32
Create New Model for Current Project Data Dictionary	command 61
Create New Project Data Dictionary	command 32, 61
Create Node	command 53
Create Relation	command 23, 50
Create Source Data Item	command 32, 61
Create Source Material Log Item	command 32, 61
Create View	command 55
Delete Activity	command 50
Delete All Projects In Data Dictionary System	command 61
Delete Attribute Class	command 55
Delete Concept	command 50, 52
Delete Entity Class	command 55
Delete Key Class	command 55
Delete Link Class	command 55
Delete Model	command 47, 55
Delete Node	command 53
Delete Relation	command 50
Describe	command 55
Edit	command 52
Edit Attribute Class Description	command 55
Edit Attribute Class in Key Class Description	command 55
Edit Comment	command 53
Edit Concept	command 23, 50
Edit Diagram Text	command 50
Edit Dictionary Item	command 61
Edit Entity Class Description	command 55
Edit Glossary Entry	command 50
Edit Inherited Attribute Class Description	command 55
Edit Key Class Description	command 55

Edit Link Class Description	command 55
Edit Model Context	command 50
Edit Model Description	command 55
Edit Model Purpose	command 50
Edit Model Viewpoint	command 50
Edit View Description	command 55
Exit	command 52
Exit Data Dictionary	command 61
Exit Editor	command 53
Hardcopy Display	command 55
Hide Comment	command 53
Hide Node Decomposition	command 53
IDEF1 Data Dictionary Editor	command 32, 47
Inspect Attribute Class	command 55
Load Model	command 47
Load Project Data Dictionary	command 61
Merge Entity Classes	command 55
Mouse-M	command 55
Move Activity	command 50
Move Entity Class	command 55
Move Link Class	command 55
Move Node	command 53
Node Editor	command 7, 50
Print Current Project and Model	command 61
Print Diagram	command 50
Print Entire Model	command 50
Print Node Tree	command 50
Print Reviewers Comments	command 55
Refresh	command 47
Relax Net	command 55
Remove Attributes From Key Class	command 55
Rename	command 52
Rename Activity	command 50
Rename Concept	command 50
Rename View	command 55
Reviewers	command 55
Reviewers Comment	command 50
Save Model	command 47
Save Project Data Dictionary	command 61
Scroll Window	command 47
Select	command 55
Select Configuration	command 47
Select Model	command 47
Select Pane	command 47
Select View	command 55
Set Select State	command 55
Show Activity Decomposition	command 53
Show Comment	command 53
Show Concept	command 50
Show Context	command 23, 50
Show Decomposition	command 50
Show Details	command 55
Show Inheritance	command 6, 52
Show Node Decomposition	command 53
Show Statistics	command 55
Swap Activities	command 50
Switch Models in the Current Project Data Dictionary	command 61
Switch Projects	command 61
Undelete Node	command 53
View Other Details of Item	command 61

Where Used	command 6, 52
	Command Completion 12
Clear	Command History command 50
IDEF0 Modeler	Command Reference 50
IDEF1 Modeler	Command Reference 55
General Model Builder	Commands 47
IDEF0 Concept Editor	Commands 52
IDEF0 Node Tree Editor	Commands 53
IDEF1 Data Dictionary Editor	Commands 61
MDSE	Commands 43
MEDD MENU	Commands 68
Summary of	Commands 77
Summary of	Commands for the MEDD 68
	Command Summary 43
	Command Window 15
Edit	Comment command 53
Hide	Comment command 53
Reviewers	Comment command 50
Show	Comment command 53
	Commenting Support 13
Print Reviewers	Comments command 55
Command	Completion 12
Create	Concept command 23, 50, 52
Delete	Concept command 50, 52
Edit	Concept command 23, 50
Rename	Concept command 50
Show	Concept command 50
IDEF0	Concept Editor 6
	Concept Editor command 23, 50
IDEF0	Concept Editor Commands 52
Creating a Model, Activities, Modifying AUTOEXC.BAT and	Concepts, and Relations 23
	CONFIG.SYS 87
Select	Configuration command 47
Guaranteed Model	Consistency 13
Change	Context command 6, 52
Edit Model	Context command 50
Show	Context command 23, 50
	Control 3
Function	Controls 66
	Conventions and Definitions 93
Model	Copy 14
	Copy Model Portion command 55
	Create Activity command 23, 50
	Create Attribute Class command 55
	Create Concept command 23, 50, 52
	Create Entity Class command 55
	Create Item for Attribute Class Pool command 32, 61
	Create Item for Entity Class Pool command 32, 61
	Create Key Class command 55
	Create Link Class command 32, 55
	Create Model command 23, 47, 55
	Create New Model Current Project Data Dictionary command 32
	Create New Model for Current Project Data Dictionary command 61
	Create New Project Data Dictionary command 32, 61
	Create Node command 53
	Create Relation command 23, 50
	Create Source Data Item command 32, 61
	Create Source Material Log Item command 32, 61

	Create View command	55
	Creating a Model, Activities, Concepts, and Relations	23
	Creating a User ID	101
Attribute Class	Cross Reference	66
Node	Cross Reference	66
Print	Current Project and Model command	61
Change Name of	Current Project Data Dictionary command	61
Create New Model	Current Project Data Dictionary command	32
Create New Model for	Current Project Data Dictionary command	61
Switch Models in the	Current Project Data Dictionary command	61
Life	Cycle Pool	66
Life	Cycle States	66

D**D****D**

Clear	Data	68
Fetch	Data	68
Print	Data	68
Project	Data Dictionary	10
Change Name of Current Project	Data Dictionary command	61
Create New Model Current Project	Data Dictionary command	32
Create New Model for Current Project	Data Dictionary command	61
Create New Project	Data Dictionary command	32, 61
Exit	Data Dictionary command	61
Load Project	Data Dictionary command	61
Save Project	Data Dictionary command	61
Switch Models in the Current Project	Data Dictionary command	61
IDEF1	Data Dictionary Editor	10
IDEF1	Data Dictionary Editor command	32, 47
IDEF1	Data Dictionary Editor Commands	61
	Data Dictionary Manager	15
Delete All Projects In	Data Dictionary System command	61
Project	Data Dictionary Tool	10
Create Source	Data Item command	32, 61
Source	Data List	10, 15, 66
Source	Data Log	32
	Date	68
Function	Decomposition	66
Show	Decomposition	23
Check	Decomposition command	50
Hide Node	Decomposition command	53
Show	Decomposition command	50
Show Activity	Decomposition command	53
Show Node	Decomposition command	53
IDEF0	Decomposition Validation	13
Attribute Class	Definition	66
Entity Class	Definition	66
Function	Definition	66
Interface	Definition	66
Relation Class	Definition	66
	Definition Oriented Forms	68
Conventions and	Definitions	93
Field Format	Definitions	68
Form Field	Definitions	68
Form Menu Item	Definitions	68
Main Menu Pick	Definitions	68
	Delete Activity command	50
	Delete All Projects In Data Dictionary System	command 61

	Delete Attribute Class command	55
	Delete Concept command	50, 52
	Delete Entity Class command	55
	Delete Key Class command	55
	Delete Link Class command	55
	Delete Model command	47, 55
	Delete Node command	53
	Delete Record	68
	Delete Record(s)	68
	Delete Relation command	50
	Describe command	55
Edit Attribute Class	Description command	55
Edit Attribute Class in Key Class	Description command	55
Edit Entity Class	Description command	55
Edit Inherited Attribute Class	Description command	55
Edit Key Class	Description command	55
Edit Link Class	Description command	55
Edit Model	Description command	55
Edit View	Description command	55
IDEF1 Tool	Design	9
Design 12	MDSE Model Development Support Requirements and	
MDSE Modeling Tools Requirements and	Design	3
Show	Details command	55
View Other dictionary	Details of Item command	61
MEDD	developer	65
MDSE Model	Development Platform	65
Model Element Dictionary	Development Support Requirements and Design	12
Model Builder	Development User's Manual	65
Print	Development Window	15
Edit	Diagram command	50
Project Data	Diagram Text command	50
Change Name of Current Project Data	Dictionary	10
Create New Model Current Project Data	Dictionary command	61
Create New Model for Current Project Data	Dictionary command	32
Create New Project Data	Dictionary command	61
Exit Data	Dictionary command	32, 61
Load Project Data	Dictionary command	61
Save Project Data	Dictionary command	61
Switch Models in the Current Project Data	Dictionary command	61
Model Element	dictionary developer	65
IDEF1 Data	Dictionary Development User's Manual	65
IDEF1 Data	Dictionary Editor	10
IDEF1 Data	Dictionary Editor command	32, 47
Edit	Dictionary Editor Commands	61
Data	Dictionary Item command	61
Delete All Projects In Data	Dictionary Manager	15
Project Data	Dictionary System command	61
Setting up	Dictionary Tool	10
Floppy	Directories	86
Hard	Directory	68
Hardcopy	Disk Installation	91
All	Disk Installation	91
Text	Display command	55
	Displayed Instances	68
	Display Window	15
	Distribution and Support of the MEDD	89
	Domains	66
	Domain Values	66

E

To DOS 68
Dynamic Windows 15

E**E**

Edit 68
Edit Attribute Class Description command 55
Edit Attribute Class in Key Class Description command 55
Edit command 52
Edit Comment command 53
Edit Concept command 23, 50
Edit Diagram Text command 50
Edit Dictionary Item command 61
Edit Entity Class Description command 55
Edit Glossary Entry command 50
ZMACS Editing Window 17
Edit Inherited Attribute Class Description command 55
Edit Key Class Description command 55
Edit Link Class Description command 55
Edit Model Context command 50
Edit Model Description command 55
Edit Model Purpose command 50
Edit Model Viewpoint command 50
Editor 6
IDEF0 Concept Editor 7
IDEF0 Node Editor 10
IDEF1 Data Dictionary Editor 13
ZMACS Editor 13
Concept Editor command 23, 50
Exit Editor command 53
IDEF1 Data Dictionary Editor command 32, 47
Node Editor command 7, 50
IDEF0 Concept Editor Commands 52
IDEF0 Node Tree Editor Commands 53
IDEF1 Data Dictionary Editor Commands 61
Edit View Description command 55
Element Dictionary Development User's Manual 65
IDEF1 Model Element Merge 14
Enterprise Model Elements 66
Function Model Elements 66
Information Model Entity Class Elements 66
Automatic Model Rule Enforcement 13
Structured English Input 12
IDEF1 English Language Statements 13
Enterprise Model 68
Enterprise Model Elements 66
Print Entire Model command 50
Entity Class 14
Back Entity Class 32
Front Entity Class 32
Create Entity Class command 55
Delete Entity Class command 55
Move Entity Class command 55
Entity Class Definition 66
Edit Entity Class Description command 55
Information Model Entity Class Elements 66
Entity Classes 8
Merge Entity Classes command 55
Entity Class Pool 10, 15, 32, 66

Create Item for Entity Class Pool command 32, 61
 Entity Class Synonyms 66
 First Time Entry 68
 Edit Glossary Entry command 50
 Exit 68
 Exit command 52
 Exit Data Dictionary command 61
 Exit Editor command 53

F**F****F**

Fact Collection Tool 73
 Micro-Based Fact Collection Tool User's Manual 73
 Keys to Become Familiar with 95
 Installation and Hardware and Software Requirements for the

FCT 91

Features of the IDEF1 Tool 9
 Fetch Data 68
 Form Field Definitions 68
 Field Format Definitions 68
 File Output of Graphics 12
 File System 20
 B+ Tree File System 65
 First Time Entry 68
 Floppy Disk Installation 91
 Form 68
 To Related Form 68
 Field Format Definitions 68
 Form Field Definitions 68
 Free Form Layout 12
 Form Menu Bar 68
 Form Menu Item Definitions 68
 Definition Oriented Forms 68
 List Oriented Forms 68
 Free Form Layout 12
 Remove Attributes From Key Class command 55
 Front Entity Class 32
 Function Controls 66
 Function Decomposition 66
 Function Definition 66
 Function Inputs 66
 Function Locations 66
 Function Mechanisms 66
 Function Model 68
 Function Model Elements 66
 Function Outputs 66
 Function Sequence 66

G**G****G**

General Model Builder Commands 47
 Generate 68
 To Getting Around in the Symbolics World 103
 Getting Started 74
 Getting Started on the MDSE 20
 Getting Started with the MEDD 66
 Glossary 8
 Edit Glossary Entry command 50
 File Output of Graphics 12
 Interactive Graphics 12

Group Locations 66
 Group Pool 66
 Guaranteed Model Consistency 13

H

Model Object Save /

IDEF0 and IDEF1

Installation and
 Installation and

Installation and

On Line

Clear Command

Clear Output

H

Hardcopy 12

Hardcopy Display command 55

Hard Disk Installation 91

Hardware and Software Requirements 88

Hardware and Software Requirements 84

Hardware and Software Requirements for the FCT 91

Hardware and Software Requirements for the MDSE
 84

Hardware and Software Requirements for the MEDD
 86

Help 68

Help 12

Hide Comment command 53

Hide Node Decomposition command 53

Hints and Assumptions of IDEF0 30

History command 50

History command 50

H

Creating a User
 Hints and Assumptions of
 Sample Usage Scenario for
 Running

The

Running IDEF0 and
 Sample Scenario for

IDEF0 and
 IDEF0 and

Features of the

Attribute Class Migration
 Technical

ID 101

IDEF0 30

IDEF0 23

IDEF0 and IDEF1 20

IDEF0 and IDEF1 Hardware and Software
 Requirements 84

IDEF0 and IDEF1 Installation Procedures 84

IDEF0 Concept Editor 6

IDEF0 Concept Editor Commands 52

IDEF0 Decomposition Validation 13

IDEF0 Modeler 3

IDEF0 Modeler Command Reference 50

IDEF0 Node Editor 7

IDEF0 Node Tree Editor Commands 53

IDEF1 20

IDEF1 32

IDEF1 Data Dictionary Editor 10

IDEF1 Data Dictionary Editor command 32, 47

IDEF1 Data Dictionary Editor Commands 61

IDEF1 English Language Statements 13

IDEF1 Hardware and Software Requirements 84

IDEF1 Installation Procedures 84

IDEF1 Model Element Merge 14

IDEF1 Modeler 8

IDEF1 Modeler Command Reference 55

IDEF1 Tool 9

IDEF1 Tool Design 9

IDEF Model Builder Status Line and Mouse Status
 Line 18

IDEF Model Builder Windows 15

Index 66

Information 92

Information Model 68

Information Model Entity Class Elements 66

Key Class to	Inherit 32
Show	Inheritance command 6, 52
Edit	Inherited Attribute Class Description command 55
	Input 3
Structured English	Input 12
Function	Inputs 66
	Insert After 68
	Insert Before 68
	Inspect Attribute Class command 55
Floppy Disk	Installation 91
Hard Disk	Installation 91
	Installation and Hardware and Software Requirements for the FCT 91
	Installation and Hardware and Software Requirements for the MDSE 84
	Installation and Hardware and Software Requirements for the MEDD 86
IDEF0 and IDEF1	Installation Procedures 84
All Displayed	Instances 68
Model	Integration Support 14
	Interactive Graphics 12
	Interface Attributes 66
	Interface Definition 66
	Introduction to Symbolics 93
	Invoking MEDD Via Q-PRO4 88
Architectural	Issues 15

K**K****K**

Add Attributes to	Key Class command 32, 55
Create	Key Class command 55
Delete	Key Class command 55
Remove Attributes From	Key Class command 55
Edit	Key Class Description command 55
Edit Attribute Class in	Key Class Description command 55
	Key Class to Inherit 32
	Keys to Become Familiar with 95
	Kit Production 13

L**L****L**

Microsoft Mouse Menu Programming	Language 65
IDEF1 English	Language Statements 13
Auto	Layout 12
Free Form	Layout 12
	Life Cycle Pool 66
	Life Cycle States 66
IDEF Model Builder Status Line and Mouse Status	Line 18
Model Builder Status	Line 18
Mouse Status	Line 18
IDEF Model Builder Status	Line and Mouse Status Line 18
On	Line Help 12
	Link Cardinality 32
Create	Link Class command 32, 55
Delete	Link Class command 55
Move	Link Class command 55
Edit	Link Class Description command 55
	Link Classes 32
Source Data	List 10, 15, 66
	List Oriented Forms 68

Loading the Model Builder 20
 Load Model command 47
 Load Project Data Dictionary command 61
 Location Pool 66
 Function Locations 66
 Group Locations 66
 Source Data Log 32
 Source Material Log 10, 15, 32, 66
 Logging In 20
 Logging In and Out 97
 Logging Out 20
 Create Source Material Log Item command 32, 61

M**M****M**

Cold-Booting the Machine 99
 Powering up the Machine 99
 Machine Activities 104
 To Main Menu 68
 Main Menu Bar 68
 Main Menu Pick Definitions 68
 Data Dictionary Manager 15
 Micro-Based Fact Collection Tool User's Manual 73
 Model Element Dictionary Development User's Manual 65
 Strong many-to-one 32
 Weak many-to-one 32
 Source Material Log 10, 15, 32, 66
 Create Source Material Log Item command 32, 61
 Getting Started on the MDSE 20
 Installation and Hardware and Software Requirements for the MDSE 84
 MDSE Commands 43
 MDSE Model Development Support Requirements and Design 12
 MDSE Modeling Tools Requirements and Design 3
 Mechanism 3
 Function Mechanisms 66
 Distribution and Support of the MEDD 89
 Getting Started with the MEDD 66
 Installation and Hardware and Software Requirements for the MEDD 86
 Summary of Commands for the MEDD 68
 MEDD Architecture 65
 MEDD Development Platform 65
 MEDD MENU Commands 68
 Invoking MEDD Via Q-PRO4 88
 To Main Menu 68
 Form Menu Bar 68
 Main Menu Bar 68
 MEDD MENU Commands 68
 Form Menu Item Definitions 68
 Main Menu Pick Definitions 68
 Microsoft Mouse Menu Programming Language 65
 IDEF1 Model Element Merge 14
 Model Merge 14
 Merge Entity Classes command 55
 Methods 68
 Micro-Based Fact Collection Tool User's Manual 73
 Microsoft Mouse Menu Programming Language 65
 Migration Index 66
 Attribute Class

Enterprise	Model 68
Function	Model 68
Information	Model 68
Save	Model 23
Creating a	Model, Activities, Concepts, and Relations 23
	Model Application / Analysis Support 15
Loading the	Model Builder 20
General	Model Builder Commands 47
	Model Builder Development Window 15
	Model Builder Status Line 18
IDEF	Model Builder Status Line and Mouse Status Line 18
IDEF	Model Builder Windows 15
Create	Model command 23, 47, 55
Delete	Model command 47, 55
Load	Model command 47
Print Current Project and	Model command 61
Print Entire	Model command 50
Save	Model command 47
Select	Model command 47
Guaranteed	Model Consistency 13
Edit	Model Context command 50
	Model Copy 14
Create New	Model Current Project Data Dictionary command 32
Edit	Model Description command 55
MDSE	Model Development Support Requirements and Design 12
	Model Element Dictionary Development User's Manual 65
IDEF1	Model Element Merge 14
Enterprise	Model Elements 66
Function	Model Elements 66
Information	Model Entity Class Elements 66
IDEF1	Modeler 8
The IDEF0	Modeler 3
IDEF0	Modeler Command Reference 50
IDEF1	Modeler Command Reference 55
Create New	Model for Current Project Data Dictionary command 61
MDSE	Modeling Tools Requirements and Design 3
	Model Integration Support 14
	Model Merge 14
	Model Object Save / Hardcopy 12
Copy	Model Portion command 55
Edit	Model Purpose command 50
Automatic	Model Rule Enforcement 13
Switch	Models in the Current Project Data Dictionary command 61
	Model Validation and Support 13
Edit	Model Viewpoint command 50
	Modifying AUTOEXC.BAT and CONFIG.SYS 87
	Modify Record 68
	Mouse-M command 55
Microsoft	Mouse Menu Programming Language 65
	Mouse Status Line 18
IDEF Model Builder Status Line and	Mouse Status Line 18
	Move Activity command 50
	Move Entity Class command 55
	Move Link Class command 55
	Move Node command 53

N

N

Change Name of Current Project Data Dictionary command 61

Relax Net command 55

Create New Model Current Project Data Dictionary command 32

Create New Model for Current Project Data Dictionary command 61

Create New Project Data Dictionary command 32, 61

Next Record 68

Move Node command 53

Create Node command 53

Delete Node command 53

Undelete Node command 53

Node Cross Reference 66

Hide Node Decomposition command 53

Show Node Decomposition command 53

IDEF0 Node Editor 7

Node Editor command 7, 50

Print Node Tree command 50

IDEF0 Node Tree Editor Commands 53

Notifications 66

O

O

Model Object Save / Hardcopy 12

One-to-one 32

View Only Window 17

Definition Oriented Forms 68

List Oriented Forms 68

View Other Details of Item command 61

Other Requirements 89

Logging Out 20

Logging In and Out 97

Output 3

To OUTPUT 68

Clear Output History command 50

File Output of Graphics 12

Function Outputs 66

output utility 65

P

P

Select Pane command 47

Parts and Subparts 23

Pathname 20

Main Menu Pick Definitions 68

MEDD Development Platform 65

Attribute Class Pool 10, 15, 32, 66

Entity Class Pool 10, 15, 32, 66

Group Pool 66

Life Cycle Pool 66

Location Pool 66

Process Pool 66

Project Pool 66

State Pool 66

Create Item for Attribute Class Pool command 32, 61

Create Item for Entity Class Pool command 32, 61

Copy Model Portion command 55

P

Powering up the Machine 99
 Presentation Types 15
 Previous Record 68
 Print Current Project and Model command 61
 Print Data 68
 Print Diagram command 50
 Print Entire Model command 50
 Print Node Tree command 50
 Print Reviewers Comments command 55
 Procedures 84
 Process Pool 66
 Production 13
 Kit Production 13
 Wall Chart
 Microsoft Mouse Menu
 Print Current
 Change Name of Current
 Create New
 Create New Model Current
 Create New Model for Current
 Load
 Save
 Switch Models in the Current
 Switch
 Delete All
 Edit Model

Project and Model command 61
 Project Data Dictionary 10
 Project Data Dictionary command 61
 Project Data Dictionary command 32, 61
 Project Data Dictionary command 32
 Project Data Dictionary command 61
 Project Data Dictionary command 61
 Project Data Dictionary command 61
 Project Data Dictionary command 61
 Project Data Dictionary Tool 10
 Project Pool 66
 Projects command 61
 Projects In Data Dictionary System command 61
 Purpose command 50

Q**Q****Q**

Invoking MEDD Via

Q-PRO 65
 Q-PRO4 88

R**R****R**

Access
 Revision
 Add
 Delete
 Modify
 Next
 Previous
 Specified
 Add
 Delete
 Attribute Class Cross
 IDEF0 Modeler Command
 IDEF1 Modeler Command
 Node Cross
 To
 Create
 Delete
 Creating a Model, Activities, Concepts, and

Reasons 66
 Reasons 66
 Record 68
 Record 68
 Record 68
 Record 68
 Record 68
 Record 68
 Record 68
 Record(s) 68
 Record(s) 68
 Reference 66
 Reference 50
 Reference 55
 Reference 66
 Refresh command 47
 Related Form 68
 Relation Class 14
 Relation Class Definition 66
 Relation Classes 8
 Relation command 23, 50
 Relation command 50
 Relations 23
 Relax Net command 55
 Remove Attributes From Key Class command 55

Hardware and Software
 IDEF0 and IDEF1 Hardware and Software
 Other
 MDSE Model Development Support
 MDSE Modeling Tools
 Installation and Hardware and Software
 Installation and Hardware and Software
 Installation and Hardware and Software
 Print
 Auto
 Automatic Model
 Rename Activity command 50
 Rename command 52
 Rename Concept command 50
 Rename View command 55
 Requirements 88
 Requirements 84
 Requirements 89
 Requirements and Design 12
 Requirements and Design 3
 Requirements for the FCT 91
 Requirements for the MDSE 84
 Requirements for the MEDD 86
 Reviewers command 55
 Reviewers Comment command 50
 Reviewers Comments command 55
 Revision Reasons 66
 Router 12
 Rule Enforcement 13
 Running IDEF0 and IDEF1 20

S

Model Object

Sample Usage
Sample

Set

Function

Hardware and
 IDEF0 and IDEF1 Hardware and
 Installation and Hardware and
 Installation and Hardware and
 Installation and Hardware and
 Create

S

Sample Scenario for IDEF1 32
 Sample Usage Scenario for IDEF0 23
 Save / Hardcopy 12
 Save Model 23
 Save Model command 47
 Save Project Data Dictionary command 61
 Scenario for IDEF0 23
 Scenario for IDEF1 32
 Scroll Window command 47
 Select command 55
 Select Configuration command 47
 Select Model command 47
 Select Pane command 47
 Select State command 55
 Select View command 55
 Sequence 66
 Sequentially 68
 Set Select State command 55
 Setting up Directories 86
 Show Activity Decomposition command 53
 Show Comment command 53
 Show Concept command 50
 Show Context command 23, 50
 Show Decomposition 23
 Show Decomposition command 50
 Show Details command 55
 Show Inheritance command 6, 52
 Show Node Decomposition command 53
 Show Statistics command 55
 Software Requirements 88
 Software Requirements 84
 Software Requirements for the FCT 91
 Software Requirements for the MDSE 84
 Software Requirements for the MEDD 86
 Source Data Item command 32, 61
 Source Data List 10, 15, 66
 Source Data Log 32
 Source Material Log 10, 15, 32, 66

S

Create	Source Material Log Item command 32, 61
	Specified Record 68
Getting	Started 74
Getting	Started on the MDSE 20
Getting	Started with the MEDD 66
Set Select	State command 55
IDEF1 English Language	Statements 13
	State Pool 66
Life Cycle	States 66
Show	Statistics command 55
IDEF Model Builder Status Line and Mouse	Status Line 18
Model Builder	Status Line 18
Mouse	Status Line 18
IDEF Model Builder	Status Line and Mouse Status Line 18
	Strong many-to-one 32
	Structured English Input 12
Parts and	Subparts 23
Types and	Subtypes 23
Command	Summary 43
	Summary of Commands 77
	Summary of Commands for the MEDD 68
Commenting	Support 13
Model Application / Analysis	Support 15
Model Integration	Support 14
Model Validation and	Support 13
Distribution and	Support of the MEDD 89
MDSE Model Development	Support Requirements and Design 12
	Swap Activities command 50
	Switch Models in the Current Project Data Dictionary
	command 61
	Switch Projects command 61
Introduction to	Symbolics 93
Getting Around in the	Symbolics World 103
Entity Class	Synonyms 66
B+ Tree File	System 65
File	System 20
Delete All Projects In Data Dictionary	System command 61

T

T

T

	Technical Information 92
The	Temporary Windows 17
Edit Diagram	Text command 50
	Text Display Window 15
	Time 68
First	Time Entry 68
	To DOS 68
	To Generate 68
	To Main Menu 68
Fact Collection	Tool 73
Features of the IDEF1	Tool 9
Project Data Dictionary	Tool 10
IDEF1	Tool Design 9
MDSE Modeling	Tools Requirements and Design 3
Micro-Based Fact Collection	Tool User's Manual 73
	To OUTPUT 68
	To Related Form 68
Print Node	Tree command 50
IDEF0 Node	Tree Editor Commands 53
B+	Tree File System 65

application Tutor 68
 Presentation tutorial 65
 Types 15
 Types and Subtypes 23

U

Setting Undelete Node command 53
 Powering up Directories 86
 Sample up the Machine 99
 Where Usage Scenario for IDEF0 23
 Creating a Used command 6, 52
 Micro-Based Fact Collection Tool User ID 101
 Model Element Dictionary Development User's Manual 73
 output User's Manual 65
 utility 65

U

U

V

IDEF0 Decomposition Validation 13
 Model Validation and Support 13
 Domain Values 66
 Invoking MEDD Via Q-PRO4 88
 Create View command 55
 Rename View command 55
 Select View command 55
 Edit View Description command 55
 Edit Model View Only Window 17
 View Other Details of Item command 61
 Viewpoint command 50

V

V

W

Command Wall Chart Production 13
 Model Builder Development Weak many-to-one 32
 Text Display Where Used command 6, 52
 View Only Window 15
 ZMACS Editing Window 15
 Scroll Window 17
 Dynamic Window command 47
 IDEF Model Builder Windows 15
 The Temporary Windows 15
 Getting Around in the Symbolics World 103

W

W

Z

Z

ZMACS Editing Window 17
 ZMACS Editor 13

Z